

JULIO 1985
Nº 4 - A 1,50.- REP. ARGENTINA
K64
COMPUTACION PARA TODOS
Copiador Para TS 2068
22 Programas
Moviendo Sprites
Menos Trabajo,
Más Juego
Concurso:
Ya está el Ganador

CZERWENY

computación para todos

Bernini & Valentini



MENOS TRABAJO, MAS JUEGO

Día a día trabajaremos menos horas, y nos va a quedar más tiempo libre, para dedicarlo a las computadoras.

En pag. **6**

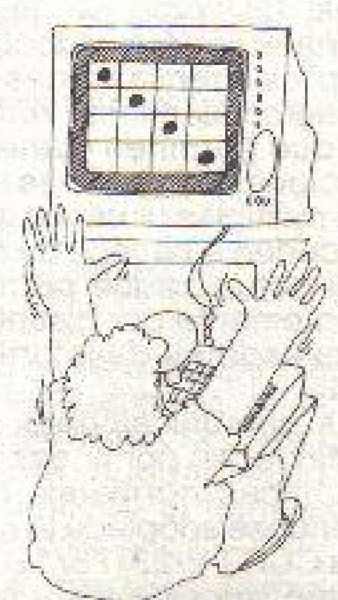
PROGRAMAS

CZ 1000/1500, TS 1000/1500, TK 83/85.
Control de cuenta bancaria (pag. 8).
Cubito, Catapulta, El Maestro, Accidente, Ta, te, ti (pag. 14 y 15).
Organito, Ecuaciones (pag. 16).
Pant (pag. 40)

CZ 2000, TK 90X, TS 2068
3D Laberinto
Diseñador de pantallas
Pesca
Gotera (pag. 42 - 46).

Commodore 64
Demo Sprite (pag. 30)
Utilitario de Sonido (pag. 31).
Rapidez de reflejos (pag. 47).
Capitales (pag. 47.).

TI 99/4A
Batalla Tierra-Aire (pag. 22 y 23)
Frontón (pag. 24)



CARTA DEL DIRECTOR

El primer Concurso de Programas K 64 llegó a su término y en este número se publican los nombres de los ganadores. Quienes tuvimos la iniciativa de hacer una revista de nivel internacional estamos verdaderamente orgullosos por la gran cantidad de trabajos recibidos y —sobre todo— por la calidad de las presentaciones, que hizo difícil la evaluación y elección final. La respuesta fue similar, y aún mejor, que la obtenida por otras publicaciones en países más avanzados en materia de informática. Esto es muy importante, porque demuestra la capacidad de nuestros cerebros para producir "software" y también para exportarlo.

Además de los premios, decidimos otorgar tres menciones.

Cuando el certamen ya estaba cerrado, recibimos muchos trabajos, algunos de ellos de gran valor, que serán incluidos, por supuesto, en el nuevo concurso, ya en marcha. Creemos que el éxito de "K 64" se basa en una relación sincera, directa y cálida con los amigos que nos leen.

Gracias por establecer esta comunicación con nosotros. Continuaremos esforzándonos por ser siempre los primeros. Y para tratar de que sea una realidad la "computación para todos", le ponemos el hombro a nuestro país, y congelamos nuestros precios al 1° de junio, mientras aumentamos la calidad de nuestro material.

CRISTIAN PUSSO

MOVIENDO SPRITES

Estudiamos varios registros del chip interfase de video VIC-II, que se encargan de manejar estos bloques móviles.

En pag. **26**

PARA DUPLICAR SOFTWARE

Un programa, íntegramente confeccionado en lenguaje de máquina, que permite realizar una lectura desde cassette almacenando en memoria el programa o bloque de datos, para a continuación volcarlo nuevamente a cassette, posibilitando de esa manera la concreción de duplicados.

En pag. **35**

SECCIONES FIJAS

Mundo informático (Pag. 4).
Trucos, trampas y hallazgos (Pag. 12).
Sorteo mensual (Pag. 34).
Conociendo las computadoras (Pag. 25).
Introducción a la computación (Pag. 48).
Correo-Consultas (Pag. 49 y 50).



COMPUTACION PARA TODOS

CORREO ARGENTINO CASA CENTRAL
FRANQUEO PAGO CONCESIÓN Nº 2530
TARIFA REDUCIDA CONCESIÓN Nº 835

AÑO 1 Nº 4 JULIO DE 1985

Director: Cristián Pusso
Director Periodistico: Fernando Flores
Jefe de Publicidad: Dolores Urien
Arte y Diagramación: Carlos Boccardo y Mario Romualdo
Secretaria: Moni Ocampo

K-64 es una Revista mensual editada por PROEDI Editorial S.A. (e./f.), Cerrito 1320, 1° Piso, Buenos Aires, Te.: 42-9681/9. Registro Nacional de la Propiedad Intelectual:313.837 M. registrada Queda hecho el dépósito que indica la Ley 11.723 de Propiedad Intelectual.

Precio de este ejemplar: un austral con cincuenta centavos. Precio de la suscripción semestral: 8 australes.
Distribuidor en Capital: Infinito. Venezuela 1417 Capital Federal. Tel.: 37-6664.
Impresión: Calcotam. Fotocromo tapa: Columbia.
Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

IBM ARGENTINA FUE LA EMPRESA QUE MAS EXPORTO DURANTE 1984

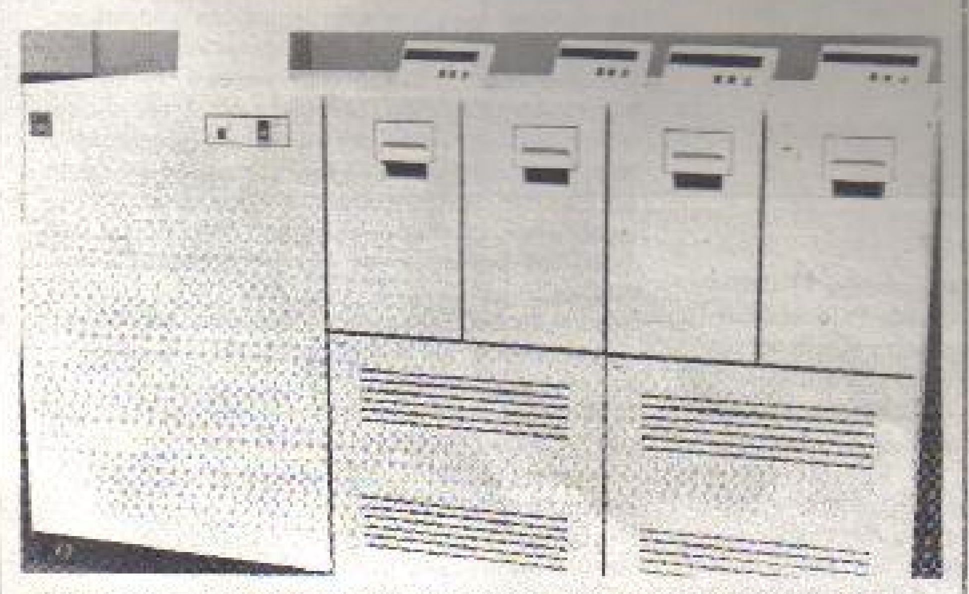
Con un volumen de exportación de más de 90 millones de dólares, IBM Argentina volvió a adjudicarse el primer premio como empresa que logró en 1984 el mayor monto neto de divisas por ese concepto, en el concurso que anualmente organiza la Asociación de Gerentes de Comercio Exterior (AGEX) y la Cámara de Comercio de los Estados Unidos de América en la República Argentina.

La distinción fue recibida por el Vicepresidente de IBM Argentina, ingeniero Julio Viau, de manos del Subsecretario de Comercio Exterior, licenciado Néstor Stancanelli, durante un almuerzo que se sirvió en las instalaciones del Club Americano. (FOTO)

El logro obtenido por IBM Argentina es una ratificación de la alta competitividad internacional que en materia de calidad y costos tiene la planta fabril de Martínez, que hoy es uno de los 16 establecimientos integradores de sistemas que la Corporación posee en el mundo entero fuera de los Estados Unidos.

Asimismo es un reconocimiento al esfuerzo realizado en inversiones, tecnología y desarrollo de proveedores locales, que permitió el año pasado anunciar la producción local del Subsistema de Cinta Magnética IBM 3480, un producto de alta tecnología electrónica, cuyo primer embarque a cliente se concretó a menos de un año de aquel anuncio. Este subsistema, junto a algunos más cuya producción comenzará en lo que resta del año, permitirá lograr en 1985 un volumen de exportación superior a los 140 millones de dólares (o sea un incremento del 55% sobre 1984).





El equipo 3480 de IBM que se exporta a varios países

PRIMER SERVICIO VIDEOTEX

Radio Victoria Informática ha presentado en nuestro medio el primer servicio **videotex**, tecnología inscripta dentro de los llamados "nuevos medios" de comunicación teleinformática.

Con este anuncio, la Argentina se incorpora al núcleo de países que disponen comercialmente de sus facilidades operativas.

VTX, tal como se denomina el servicio videotex, implica intensas redes de teleprocesamiento remoto que permiten comunicaciones interactivas entre múltiples usuarios que acceden a bancos de datos actualizados permanentemente por distintos proveedores de información.

Las comunicaciones se establecen por líneas telefónicas normales, habiéndose adoptado la norma CCITT V-23 (75/1200) como velocidades y modo de transmisión, lo que garantiza excelentes prestaciones en la red nacional de ENTel. Simultáneamente y con transparencia total para el usuario, la red instalada prevé el intenso uso de la red ARPAC y de las líneas destinadas a comunicaciones troncales.

Los servicios incorporan la transmisión telefónica en gráficos de todo tipo (publicitarios, demostrativos, logotipos, etc.) presentables en más de 4000 colores utilizando el protocolo norteamericano-canadiense de nivel de presentación NAPLPS, estándar alfageométrico ultracompacto de máxima eficiencia en las redes telefónicas argentinas.

El acceso a las facilidades gráficas será posible a través de terminales de muy bajo costo Electrohome EGT-100 equipadas con monitores Hitachi 16" o mediante computadores personales (PC) marca IBM, convenientemente configurados.

El software de aplicación de videotex instalado en el computador central de Radio Victoria Informática es de origen norteamericano y permite el acceso tanto a las terminales mencionadas como así también a otras que soporten niveles de presentación menores, tales como teletel y ASCII.

Dentro de los múltiples servicios incluidos en VTX se destacan los agropecuarios, turísticos, jurídicos, inmobiliarios, publicitarios, bancarios, médico asistenciales, para tarjetas de crédito, de uso para grupos cerrados (empresas, cámaras, asociaciones, etc.), así como muchos otros establecidos "a medida".

Cabe destacarse también



que Radio Victoria Informática dispone de la tecnología completa de videotex "llave en mano" para grandes instalaciones privadas y que dentro de sus metas está prevista la comercialización y soporte tecnológico para otros países latinoamericanos.

Merece especial atención en relación a este lanzamiento que la Argentina es el cuarto país del mundo, después de Canadá, Estados Unidos y Japón, en implementar una red de videotex con los protocolos más modernos en la materia.

Esta empresa totalmente nacional, que desarrolla la red de teleinformática más moderna de América Latina forma parte del Grupo Empresario Argentino Radio Victoria, el cual desarrolla y comercializa productos y servicios electrónicos en el país desde hace treinta y siete años, incluyendo desde 1970, la fabricación bajo licencia de televisores y equipos Hitachi.

educativas, oficiales y privadas, y cursos intensivos de lenguaje y programación Basic que se dictará durante las vacaciones de invierno, época que se presta para un aprendizaje acelerado sin que se superponga con las actividades tradicionales de la época escolar.

A esto se suma un plan muy particular para colegios privados (momentáneamente, ya que en un futuro se hará extensivo a colegios municipales) e instituciones deportivas, por el cual EPI instalará aulas totalmente equipadas para la enseñanza de lenguaje Logo y Basic.

Estos planes se han podido lograr gracias al aporte desinteresado de importantes empresas que están contribuyendo con los objetivos fijados por EPI.

Para solicitar mayor información dirigirse a EPI Empresa para Informática, Suipacha 946 Capital Código Postal 1055 ó telefónicamente al 311-8618/49-7985/46-2529/46-7877/392-6816/20.

JORNADAS SOBRE EDUCACION Y COMPUTACION

Durante el próximo receso escolar de vacaciones de invierno se realizarán en esta Capital las primeras jornadas de "Educación y Computación" organizadas por EPI Empresa Para Informática.

Durante el desarrollo de las mismas, destacados especialistas disertarán sobre el tema Educación y Computación en particular, aportando sus diferentes experiencias.

Funcionará además un centro de información y muestra de elementos de computación para educación que abarcará todos los rubros (hardware, software, amoblamiento, bibliografía, etc.).

Este evento contará con la participación de empresas proveedoras del rubro y será auspiciado por instituciones oficiales y privadas, conscientes de la importancia que ha adquirido el tema en nuestro país.

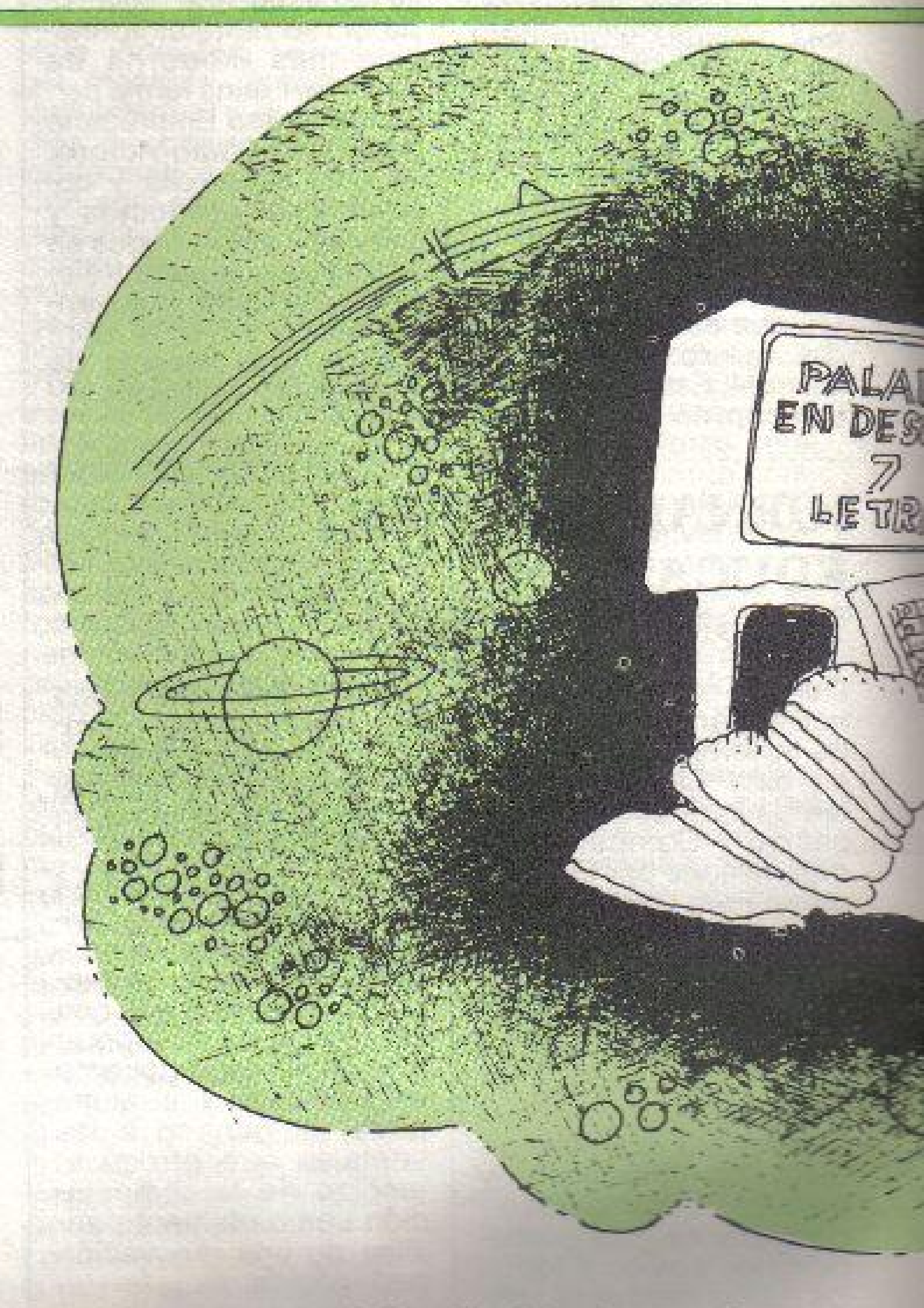
Estas jornadas forman parte del plan de actividades que ha implementado EPI para el segundo semestre del año con el objetivo de incorporar a la comunidad docente en particular y al alumnado en general a las ventajas que ofrece el empleo de la computación como elemento auxiliar de una renovación pedagógica.

EPI ha implementado además un sistema de becas destinado especialmente a educadores, pedagogos y autoridades



MENOS TRABAJO, MAS JUEGO

Hace más de 2000 años, Aristóteles hizo el comentario de que nosotros trabajamos para obtener ocio. Para él, el ocio fue una realidad ya que la gran parte del trabajo de esa época era hecho por esclavos. Pero para mucha más gente el trabajo ha sido, y es, la parte esencial en la vida de un ser humano. El ocio y la distracción sólo han tomado fracciones reducidas de su tiempo.
Hoy en día con la introducción y desarrollo de las microcomputadoras estamos produciendo un nuevo tipo de esclavo, con la forma de robots electrónicos de todo tipo. Esto representa un cambio importante en nuestras vidas ya que dará un nuevo significado y una nueva dimensión al trabajo y al ocio.
Tal vez lo que realmente significa es que la nueva tecnología hará crecer la cantidad de tiempo libre, la gente necesitará trabajar menos y deberá aprender a desarrollar su vida del mismo modo que en la época de Aristóteles.
En alguna medida, ya estamos en los comienzos de esta nueva era. Comenzamos tal vez en la década del 70 cuando la cantidad de tiempo promedio de un trabajador full-time a lo largo de un año cayó por debajo del tiempo libre disponible, en los países desarrollados.



Desde entonces un trabajador tipo ha aumentado su tiempo libre en un 7%, hasta más de 2500 horas al año mientras que las horas trabajadas cayeron a 1950, incluyendo viajes hacia y desde el lugar de trabajo.
Menos positivo es el resultado que ahora tenemos con una gran cantidad de tiempo libre forzado y mal distribuido en los millones de desocupados.
Con la expansión del tiempo libre y la necesidad de hallar el modo de ocuparlo de una manera satisfactoria, el desarrollo de la microcomputadora con su cualidad de atrapar al usuario es interesante. Ciertamente los home computers se convertirán poco a poco en lo más importante de la vida de la gente (más de como lo es hoy en día el fenómeno de la televisión). Pero necesitamos ser realistas en evaluar qué tan rápido se desarrollarán los usuarios particulares de estas micros, y también cómo llegarán a ser usadas estas maquinitas.
Sin profundizar demasiado en la psicología humana respecto del ocio, es útil tratar de delinear un poco los posibles roles que jugarán las microcomputadoras en nuestros ratos libres, estudiando en qué empleamos nuestro ocio.
Es posible identificar tres funciones principales del ocio, que forman cierta jerarquía:

- Descanso y recuperación del trabajo.
- Entretenimiento y alivio del aburrimiento.
- Y finalmente, desarrollo personal y social.

en la era en que estamos entrando, la función del ocio como descanso está declinando en importancia debido a que la cantidad y esfuerzo del trabajo decrece también. Esto hace que aumente el énfasis en las otras dos áreas (entretenimiento y desarrollo personal) y que sea allí donde dentro de poco se dirigirá el uso de la computadora en nuestro tiempo libre.
Donde más se destaca por su popularidad, es en el área del entretenimiento, donde la computadora se convierte en virtualmente, un compendio infinito de juegos. Ella puede proveernos de un sinfín de pasatiempos para todos aquellos que querramos ocupar nuestro tiempo libre de esé modo.
Podría argumentarse que bien puede usarse también un simple mazo de cartas o un cubilete, o un tablero de ajedrez, pero la respuesta es-

En promedio, día a día trabajamos menos horas al año y nos va quedando más tiempo libre. Las computadoras tienen algo que ver en ésto y también en cómo llenaremos las horas con ellas en el futuro.



tá en la atracción de usar una nueva tecnología, como así también el perpetuo desafío de los juegos de computadora, además de la amplia variedad de pasatiempos disponibles.

Una vez atrapado por la computadora, el jugador-maníaco tiende a buscar nuevos juegos más complejos y más desafiantes.

Eventualmente se lanzará al perfeccionamiento o desarrollo propio de juegos, convirtiéndose en un hobby que le tomará mucho tiempo pero que quizá será muy redituable también.

Mucha gente se volcó a las microcomputadoras sin demasiado énfasis en los juegos. Así esté su interés en el hardware o en el software, todos los hobbistas de la computación hallaron una ocupación que les absorbe.

En el futuro, cuando haya más tiempo libre aún, pero probablemente no demasiado dinero extra para gastar en ocio, esta característica atrapante de la computadora será valiosa.

Además, como bien lo saben los que forman grupos o clubes de usuarios, se puede hallar un nuevo y fuerte elemento social en este hobby, logrando nuevos compañeros, como así también un útil intercambio de experiencias y de ideas.

Otro de los aspectos que mencionábamos es el del auto-desarrollo personal, donde es de esperar que la microcomputadora juegue un rol fundamental.

Por auto-desarrollo queremos significar al proceso completo de aprendizaje y mejoramiento de las capacidades físicas e intelectuales, con el objetivo de lograr una vida más completa y rica.

Naturalmente una computadora no hará jogging por nosotros, pero sí será una herramienta de ayuda en todas las áreas del desarrollo cultural. Antes y aún hoy en día sólo podían asistir a las universidades aquéllos que disponían de tiempo libre para los cursos y así poder luego disfrutar de los beneficios del aprendizaje superior. Pronto esto estará al alcance de todos en su propio hogar.

¿Qué tan pronto sucederá ésto, qué tan pronto tendremos todos una computadora en casa como aseguran los futuristas, cuánta gente se animará a usar estos aparatos extraños para educarse a sí mismos o a su familia?

Se cree que a fines de esta década, por lo menos uno de cada 50 hogares (unos 4 millones) habrán comprado una de estas maquinitas para uso personal; otros, como ahora, tendrán máquinas para usarlas, tanto en su trabajo como en su casa.

Desde este punto de vista es de esperar una gran caída en los precios, a un tercio de lo que valen ahora. Pero hay un 33% de hogares donde la cabeza de familia es mayor de 60 años, quienes ni aún costando menos de 10 dólares comprarían una computadora.

A pesar del aumento del tiempo libre, aún hay mucha gente que no tiene muy claro para qué le podría servir una computadora personal. La atracción por los juegos es de momento la principal y obvia, y la demanda para uso educativo está dirigida principalmente a los más chicos.

Tal vez para 1990 la primera generación de chicos que usan la computadora como algo de todos los días, llegue a adultos. Entonces se dará una real popularización de la computadora para uso educacional a todo nivel.

Seguramente Aristóteles se hubiera maravillado y disfrutaría el desafío de la computación, pero aún él, hubiera necesitado un poco de coraje para dar los primeros pasos en esta nueva dirección del ocio.

CONTROL DE CUENTA BANCARIA

COMP: CZ1000/1500 TK83/85
CONF: 16 K
CLAS: COM

Este programa le permitirá llevar el registro de movimientos bancarios de su cuenta, con todo detalle, en base a la definición de códigos de operación (por ejemplo: código para débitos, pagos, etc.).
Una vez cargado el programa, éste se inicializa automáticamente, mostrando un menú de opciones que detallaremos más adelante.
Los pasos a seguir para la conservación de sus datos son los siguientes: Una vez cargado el programa y lleno con sus datos particulares, para conservarlos, deberá grabar en otro cassette, manteniendo de esta manera el programa original para darle uso en otra de sus cuentas bancarias, en otro banco, etc.
Una vez inicializado el programa, Ud. verá en pantalla el siguiente menú:

0-INICIO
1-MOVIM.
2-CONSULTA
3-GASTOS
4-GRABA
5-INGRESOS
6-TABLAS

Por supuesto lo primero que Ud. debe hacer es: inicializar el programa, para ello pulse la tecla "0". A continuación deberá responder a la pregunta de NUEVO PERIODO? (SI/NO). Se refiere a que si desea comenzar un nuevo período de cuentas, borrando los datos anteriores o desea continuar con los datos ya almacenados si los hubiere.
En caso de responder afirmativamente, pulse la tecla S y luego deberá indicar el rótulo de identificación, correspondiente al banco para el cual se desea llevar el registro de cuenta.
Luego a continuación deberá entrar la fecha como año, dos dígitos seguido de ENTER, y el mes (p. ej. en caso de ser mes 1 entrar 01), y el saldo.
Luego de un momento verá el menú completo con los datos entrados recientemente. Ahora está listo para trabajar.
Este programa trabaja con códigos de operación. Para poder acceder a ellos, pulse la tecla 6, se pondrá en modo inverso y aparecerá un sub-menú con las siguientes opciones:

PANTALLA




```
DIA CODIGO      DEBE        HABER
12- 01   $A     2100
21- 61   $A                  1000
23- 62   $A               1000000
23- 02   $A    21345
28- 03   $A    12344
28- 04   $A    12333
29- 05   $A    12000
30- 06   $A     1200

1-MENU  2-CONT  3-COPY  4-SAL
```

```
                  AL 30/07/85
SEGBA       $A     2100     3.42
COLEGIO     $A    21345     34.8
CLUB        $A    12344    20.12
VESTIMENTA  $A    12333    20.11
IMPUESTOS   $A    12000    19.56
ENTEL       $A     1200     1.95

            $A

  1-MENU              3-CONT.
```

```
01-SEGBA            02-COLEGIO
03-CLUB             04-VESTIMENTA
05-IMPUESTOS        06-ENTEL
07-ALIMENTOS        08-VARIOS
09-----------       10-----------
11-----------       12-----------
13-----------       14-----------
15-----------       16-----------
17-----------       18-----------
19-----------       20-----------
21-----------       22-----------
23-----------       24-----------
25-----------       26-----------
27-----------       28-----------
29-----------       30-----------

1-MENU  2-A/B/M            4-CONT.
```

1-Cod. Egr.: se refiere a los códigos disponibles para indicar los egresos. Luego de pulsar "1", verá la tabla de códigos disponibles ya desde el 01 hasta el 30. Al pie de página, hay otras 4 opciones:
1-MENU: retorno al menú principal.
2-A/B/M: para definir cada uno de los códigos.
3-COPY: hace una copia en impresora de esta página.
4-CONT: muestra el resto de los códigos previstos para egresos (va del 31 al 60, manteniendo la misma mecánica de operación que los 30 primeros.
Para definir un código se opera de la siguiente manera:
1) Pulsar la opción 6 del menú principal.
2) Pulsar la opción 1 del sub-menú del pie de página.
3) Una vez con la lista de códigos por egresos en pantalla, pulsar la opción 2.
4) Entre el Nº bajo el cual quedará definido el cargo, p. ej. GAS, al cual le daremos el código 09. Luego en la posición correspondiente a la 09, se imprimirá un "?", y el computador le preguntará el detalle del cargo, teclee entonces GAS seguido de ENTER. Verá que el cargo GAS, está definido en el código por egresos 09. Si desea definir otro cargo responda con un SI a la pregunta de otro cambio? y si ha terminado respondo con NO, por lo que el computador volverá al menú principal.
2-Cod. Ing.: Se refiere a los ingresos que Ud. disponga, y deberá codificarlos igual que en el caso anterior.
3-Cod. Deb.: Se refiere a los códigos por débitos, y responden a la misma mecánica anteriormente vista.

Ya tenemos entonces definidos los items de operación, egresos, ingresos y débitos.

Para registrar un movimiento dado, pulse la opción 1 en el menú principal, a continuación se verá en pantalla una ficha sin datos y el saldo. Un signo ? le indica el orden de entrada de datos.

1) Entre el día de realizado el movimiento p. ej.: 02.

2) Entre el código que indica el tipo de movimiento:

1=PAGO
2=DEPOSITO
3=DEBITO

3) Entre el Nº de cheque.

4) Entre el destino y monto del cheque en cuestión (en su respectivo orden, y cuando el computador se lo pregunte).

5) Entre el código de egresos correspondiente a este movimiento, por ejemplo pago.

6) Si los datos son correctos, responda SI, por lo que este movimiento queda ya registrado. Para los movimientos restantes: DEPOSITO y DEBITO, la mecánica es idéntica, sólo debe remitirse a responder las preguntas del computador.

La opción siguiente en el menú principal, el ítem CONSULTA, se accede pulsando el 2, a lo que el computador preguntará: desde qué día?; entre pues el día desde el que desea tener una consulta al archivo y luego seleccione una de las opciones presentadas. Si opta por la 1-LISTADO, obtendrá una planilla de todos los movimientos registrados desde la fecha en que se le pidió. Pudiendo realizar una copia a impresora, pulsando el 3. Retorna al menú principal con la opción 1.

Si prefiere puede hacer una consulta de movimientos, que dará una a una las fichas del archivo desde la fecha pedida. Las opciones que aparecen al pie de página, le permitirán volver al menú principal, ver el próximo movimiento o copiar en impresora la ficha visualizada.

La opción 3 del menú principal le mostrará una planilla con el detalle de todos los gastos registrados hasta el momento. Puede retornar al menú principal, hacer una copia o continuar con el análisis.

La opción 4 del menú principal le permite grabar el programa con los datos conservando de esta manera el archivo en cinta.

La opción 5 le permite analizar los ingresos registrados, por lo que se visualiza en pantalla una planilla con dichos ingresos.

Por último la opción 6 le permite ver las tablas de códigos de operación detallados anteriormente. También puede consultar esta opción a modo de ayuda memoria, cuando no tenga presente estos códigos.

```
   1 REM ********************
   2 REM * CUENTA BANCARIA *
   3 REM ********************
   4 REM
  90 GOSUB 500
  95 GOSUB 700
  96 GOSUB 1900
  97 IF A$="5" AND L2<>X THEN GO
TO 9000
  98 IF A$<"0" OR A$>"5" THEN GO
TO 96
 100 IF A$>"0" AND A$<"5" AND L
2=X THEN GOTO 96
 104 IF L3=P1 THEN GOTO 120
 110 GOTO (VAL A$+1)*1000
 120 IF A$="2" OR A$="3" OR A$="
5" THEN GOTO 110
 130 GOTO 96
 500 FAST
 501 LET Y=21
 502 LET X=0
 503 LET P1=1
 504 LET P2=2
 505 LET P3=3
 506 LET P4=6
 507 LET P5=9
 508 LET P6=17
 509 LET P7=18
 510 LET P8=19
 511 LET C=X
 512 LET U$="85"
 513 LET L2=P1
 514 DIM H$(P3,P3+P1)
 515 LET H$(P1, TO )="PAGO"
 516 LET H$(P2, TO )="DEP."
 517 LET H$(P3, TO )="DEB."
 518 LET M$="  "
 519 LET F$="00"
 520 LET B=90
 522 LET X$="
"
 523 LET Y$="--------------------------
---------------"
 524 LET E$="ENTRE "
 525 LET SS=X
 526 DIM Z(B)
 527 LET SA=X
 528 LET ZX=75
 529 LET SO=X
 530 LET R$=" (SI/NO)"
 531 LET L2=X
 532 LET T$="DATOS"
 533 LET L3=X
 534 LET L1=X
 535 LET P9=20
 536 LET R=11
 537 LET SAL1=X
 538 LET R1=30
 539 DIM P(ZX)
 541 DIM Z$(B,R1)
 542 DIM B$(ZX,R)
 543 FOR J=P1 TO ZX
 544 LET P(J)=X
 545 LET B$(J, TO )=Y$( TO R)
 546 NEXT J
 547 DIM D(R+P1)
 548 FOR J=P1 TO B
 549 LET Z$(J, TO )=X$( TO R1)
 550 LET Z(J)=X
 551 NEXT J
 552 LET Q1=22
 553 LET Q2=R1
 554 LET SE=X
 555 LET SD=X
 557 LET Z$(P1,P3 TO P4)="000183
"
 562 LET M$="00"
 563 LET S$="ROTULO"
 564 DIM Q$(P3,Y)
 565 LET Q$(1, TO )="CARGOS POR [illegible]"
 566 LET Q$(2, TO )="CARGOS POR [illegible]"
 567 LET Q$(3, TO )="CARGOS POR [illegible]"
 571 LET D(P1)=R1+P1
 573 LET D(P3)=D(P1)
 574 LET D(P3+P1)=R1
 575 LET D(P2+P3)=D(P1)
 576 LET D(P3+P3)=R1
 577 LET D(P4-P1)=D(P1)
 578 LET D(P4)=D(P1)
 579 LET D(P5)=R1
 580 LET D(P5+P1)=D(P1)
 581 LET D(R)=R1
 582 LET D(R+P1)=D(P1)
 584 SLOW
 590 RETURN
 594 IF VAL U$/4-INT (VAL U$/4)=
0 THEN GOTO 597
 595 LET D(P2)=R1-P2
 596 GOTO 598
 597 LET D(P2)=R1-P1
 599 RETURN
 600 PRINT AT Y,X;X$
 605 RETURN
 700 PRINT AT X,P2;X$( TO 17);AT
 X,P2;S$;AT X,Y;F$;"/";M$;"/";U$
;AT P1,X;Y$;AT P2,P1;"0-INICIO
1-MOVIM.   2-CONSULTA";AT P3,P1;"
3-GASTOS  4-GRABAR   5-INGRESOS "
;AT P3+P1,P1;"6-TABLAS";AT P2+P8
,X;Y$
 705 RETURN
1000 PRINT AT P2,P1;"0";AT Y,X;"
NUEVO PERIODO?";R$
1005 LET A$=INKEY$
1010 IF A$="N" THEN GOTO 1025
1015 IF A$="S" THEN GOTO 1100
1020 GOTO 1005
1025 GOSUB 600
1030 PRINT AT Y,X;"NUEVO MES?";R
$
1035 LET A$=INKEY$
1040 IF A$="S" THEN GOTO 1062
1045 IF A$="N" THEN GOTO 1055
1050 GOTO 1035
1055 GOSUB 600
1058 LET L2=P1
1059 GOTO 95
1060 GOSUB 600
1065 PRINT AT Y,X;E$;"MES NUMERO
"
1068 INPUT M$
1070 IF M$<"01" OR M$>"12" OR LE
N M$<>2 THEN GOTO 1068
1071 IF M$<"01" OR M$>"12" OR LE
N M$<>2 THEN GOTO 1068
1072 LET F$="00"
1073 LET MES=VAL M$
1074 LET SO=X
1075 IF SAL1<>X THEN GOTO 1080
1076 GOSUB 600
1077 PRINT AT Y,X;E$;"SALDO"
1078 INPUT SAL1
1080 GOSUB 600
1084 GOSUB 1700
1086 GOTO 1055
1090 LET L1=X
1098 GOTO 1084
1100 GOSUB 600
1105 PRINT AT Y,X;E$;"ROTULO DE
IDENTIFICACION"
1107 INPUT S$
1108 IF LEN S$>17 THEN GOTO 1107
1109 GOSUB 600
1110 PRINT AT Y,X;E$;"ANIO"
1115 INPUT U$
1117 IF U$="" THEN GOTO 1057
1118 IF U$<"83" OR U$>"99" OR LE
N U$<>2 THEN GOTO 1115
1120 GOSUB 594
1126 LET L2=P1
1128 LET SAL1=X
1130 GOTO 1060
1700 FAST
1710 FOR J=P2 TO C
1720 LET Z$(J, TO )=X$( TO R1)
1730 LET Z(J)=X
1740 NEXT J
1760 LET C=P1
1762 FOR J=P1 TO ZX
1763 LET P(J)=X
1764 NEXT J
1765 LET Z$(P1,P3 TO P8)=F$+M$+U
$
1766 LET SS=X
1767 LET SE=X
1768 LET SD=X
1770 SLOW
1780 RETURN
1900 LET A$=INKEY$
1905 IF A$="" THEN GOTO 1900
1910 RETURN
2000 PRINT AT P2,P5+P2;"■"
2005 LET C=C+1
2010 IF C>B THEN GOTO 2400
2012 LET L$=F$
2015 GOSUB 2500
2017 PRINT AT P5,P1;"SALDO $A "
INT (SAL1*100)/100
2020 GOSUB 2000
2025 IF A$<>"" THEN GOTO 2050
2030 LET Z$(C, TO )=X$( TO Q2)
2032 LET Z(C)=X
2033 LET C=C-P1
2034 LET F$=L$
2035 GOSUB 600
2040 GOSUB 2600
2042 PRINT AT P5,X;X$
2045 GOTO 95
2050 LET Z$(C,P3 TO P4)=F$+M$+U$
2051 PRINT AT Y,X;E$;T$
2052 PRINT AT P5,Y+P1;"?"
2055 INPUT A$
2057 IF A$="" THEN GOTO 2030
2060 IF A$<"1" OR A$>"3" OR LEN
A$<>1 THEN GOTO 2055
2063 PRINT AT P5,Y+P1;H$(VAL A$)
2064 LET Z$(C, TO P1)=A$
2065 LET W$=Z$(C, TO P1)
2066 IF A$>"2" THEN GOTO 2120
2070 PRINT AT R,R+P3;"?"
2075 INPUT A$
2076 IF A$="" AND Z$(C, TO P1)="
2" THEN GOTO 2120
2080 IF A$="" THEN GOTO 2030
2085 IF LEN A$>P5 THEN GOTO 2075
2090 LET Z$(C,P6+P1-LEN A$ TO P6
)=A$
2095 PRINT AT R,R+P3;A$
2100 PRINT AT R+P2,P5;"?"
2105 INPUT A$
2106 IF A$="" THEN GOTO 2030
2107 IF LEN A$>P5 THEN GOTO 2105
2110 PRINT AT R+P2,P5;A$
2115 LET Z$(C,Q1 TO Q2)=A$
2120 PRINT AT P6,P6-P1;"?"
2125 INPUT A$
2130 IF A$="" THEN GOTO 2030
2135 IF A$<"0.01" OR A$>"9999999
99" THEN GOTO 2125
2140 LET Z(C)=VAL A$
2142 PRINT AT P6,P6-P1;Z(C)
2145 PRINT AT P6-P2,P5;"?"
2150 INPUT A$
2151 IF A$="" THEN GOTO 2030
2152 IF A$<"01" OR A$>"75" OR LE
N A$<>2 THEN GOTO 2450
2153 IF W$="1" AND A$>"60" THEN
GOTO 2150
2154 IF W$="3" AND (A$<"71" OR A
$>"75") THEN GOTO 2150
2155 IF W$="2" AND (A$<"61" OR A
```

K64

```
$>"70") THEN GOTO 2150
2156 PRINT AT P6-P2,P5;B$(VAL A$
, TO )
2157 LET Z$(C,P7 TO P8)=A$
2160 PRINT AT Y,X;"DATOS OK?";R$
2165 LET A$=INKEY$
2170 IF A$="S" THEN GOTO 2300
2175 IF A$="N" THEN GOTO 2205
2180 GOTO 2165
2205 GOSUB 600
2210 GOSUB 2600
2215 GOTO 2015
2300 LET P(VAL Z$(C,P7 TO P8))=P
(VAL Z$(C,P7 TO P8))+Z(C)
2301 IF W$>"1" THEN GOTO 2350
2302 LET SS=SS+Z(C)
2312 LET SAL1=SAL1-Z(C)
2315 GOTO 2357
2350 IF W$="3" THEN GOTO 2365
2352 LET SE=SE+Z(C)
2355 LET SAL1=SAL1+Z(C)
2357 PRINT AT P6,P1;X$;AT P8,P1;
"SALDO $A ";INT (SAL1*100)/100
2360 GOTO 2035
2365 LET SD=SD+Z(C)
2367 GOTO 2312
2400 PRINT AT Y,X;"NO HAY * LUGA
R. BORRE Y CONTINUE"
2405 LET L3=P1
2410 GOSUB 1900
2448 PRINT AT P2,R;"1"
2449 GOTO 5000
2450 LET LLL=VAL U$
2455 IF W$>"1" THEN GOTO 2459
2456 LET M1=P1
2457 LET M2=30
2458 GOTO 2465
2459 IF W$>"2" THEN GOTO 2463
2460 LET M1=61
2461 LET M2=70
2462 GOTO 2464
2463 LET M1=71
2464 LET M2=75
2465 LET I=C
2466 CLS
2467 PRINT TAB 8;G$(LLL, TO ),
2468 GOSUB 9400
2469 PRINT AT Y;X;" PRESIONE ENT
ER P/CONTINUAR"
2485 LET A$=INKEY$
2486 IF CODE A$<>118 THEN GOTO 2
485
2487 IF M2=30 THEN GOTO 2900
2490 CLS
2495 GOSUB 700
2496 GOSUB 2500
2497 GOSUB 2700
2498 GOTO 2145
2500 PRINT AT P5,P1;"FECHA   /
/";AT P5,P5;"DIA";AT R,P1;"CHEQ
UE NRO.";AT R+P2,P1;"DESTI.";AT P
6-P2,P2;"CARGO";AT P8,P5;"MONTO
$"
2510 RETURN
2600 FOR K=9 TO 17 STEP 2
2605 PRINT AT K,X;X$
2610 NEXT K
2615 RETURN
2700 PRINT AT P5,P4-P1;Z$(I,P3 T
O P3+P1);AT P5,P5+P1;Z$(I,P3+P2
TO P3+P3);AT P5,R+P2;Z$(I,P4-P1
TO P4)
2705 IF Z$(I, TO P1)=" " THEN GO
TO 2720
2710 PRINT AT P5,Y+P1;H$(VAL Z$(
I, TO P1))
2720 PRINT AT R,R+P3;Z$(I,P5 TO
P6)
2725 IF Z$(I,Q1 TO Q2)=X$( TO P5
) THEN GOTO 2735
2730 PRINT AT R+P2,P5;Z$(I,Q1 TO
 Q2)
2735 IF Z$(I,P7 TO P8)="  " THEN
 GOTO 2745
2740 PRINT AT P6-P2,P5;B$(VAL Z$
(I,P7 TO P8), TO )
2745 IF Z(I)=X THEN GOTO 2760
2750 PRINT AT P8,P6-P1;Z(I)
2760 RETURN
2800 PRINT AT P5,P4-P1;"?";AT Y,
X;E$;"DIA"
2810 INPUT A$
2812 IF A$="" THEN GOTO 2870
2815 IF A$<"01" OR A$>"31" OR LE
N A$<>P2 OR VAL A$>D(VAL M$) OR
A$<F$ THEN GOTO 2810
2820 LET F$=A$
2822 GOSUB 600
2825 PRINT AT P5,P5-P2;A$;AT P5,
P5+P1;M$;AT P5,R+P2;U$
2870 RETURN
2900 LET M1=31
2905 LET M2=60
2910 GOTO 2466
3000 PRINT AT P2,Y;"■";AT Y,X;"D
ESDE QUE DIA?"
3005 INPUT A$
3010 IF A$="" THEN GOTO 3070
3015 IF A$<"01" OR A$>STR$ D(MES
) OR LEN A$<>2 THEN GOTO 3005
3016 PRINT AT 21,X;"    1-LISTADO
    2-MOVIMIENTO"
3017 LET I$=INKEY$
3018 IF I$<"1" OR I$>"2" THEN GO
TO 3017
3019 IF I$="1" THEN GOTO 3800
3020 GOSUB 600
3025 FOR J=P1 TO C
3030 IF Z$(J,P3 TO P3+P1)<A$ THE
N GOTO 3065
3035 GOSUB 2500
3040 LET I=J
3045 GOSUB 2700
3050 PRINT AT Y,P2;"1-MENU  2-PR
OX.MOV.  3-COPY"
3055 LET I$=INKEY$
3056 IF I$="3" THEN COPY
3057 IF I$<"1" OR I$>"2" THEN GO
TO 3055
3058 GOSUB 600
3059 GOSUB 2600
3060 IF I$="1" THEN LET J=C
3065 NEXT J
3070 GOSUB 600
3075 GOTO 95
3500 PRINT "DIA CODIGO    DEBE
    HABER"
3510 PRINT Y$
3515 RETURN
3700 PRINT AT Y,NOT PI;"1-MENU
2-CONT  3-COPY  4-SALDO"
3705 RETURN
3800 REM
3820 CLS
3821 LET K1=X
3822 GOSUB 3500
3825 FOR J=P1 TO C
3830 IF Z$(J,P3 TO P3+P1)<A$ THE
N GOTO 3850
3835 LET K1=K1+P1
3840 IF K1/P8-INT (K1/P8)=X THEN
 GOTO 3851
3842 IF VAL (Z$(J,P7 TO P8))>=61
 THEN PRINT Z$(J,P3 TO P3+P1);"-
";Z$(J,P7 TO P8);"  $A ";TAB 29
-LEN STR$ INT Z(J);Z(J)
3843 IF VAL (Z$(J,P7 TO P8))<=60
 THEN PRINT Z$(J,P3 TO P3+P1);"-
";Z$(J,P7 TO P8);"  $A ";TAB 16
-LEN STR$ INT Z(J);Z(J)
3850 NEXT J
3851 GOSUB 3700
3852 LET I$=INKEY$
3853 IF I$<"1" OR I$>"4" THEN GO
TO 3852
3854 IF I$="4" THEN GOTO 9800
3855 IF I$="3" THEN COPY
3856 IF I$="2" THEN GOTO 3862
3857 IF J=C+P1 THEN GOTO 3860
3858 LET J=C
3859 GOTO 3850
3860 CLS
3861 GOTO 95
3862 IF J=C+P1 THEN GOTO 3860
3863 CLS
3864 GOSUB 3500
3865 GOTO 3842
4000 PRINT AT P3,P1;"■"
4001 LET UU=X
4002 CLS
4003 LET M1=P1
4004 LET M2=15
4005 CLS
4006 PRINT " ANALISIS DE GASTOS
AL";AT X,Y+P2;F$;"/";M$;"/";U$;A
T P1,X,Y$
4015 LET LL=P1
4020 FOR J=M1 TO M2
4025 IF P(J)=X OR B$(J, TO )=Y$(
 TO R) THEN GOTO 4100
4027 LET LL=LL+P1
4028 IF UU=P1 THEN GOTO 6500
4030 PRINT B$(J, TO );"$A";AT LL
,Y-LEN STR$ INT P(J);P(J);AT LL,
28-LEN STR$ INT ((P(J)*10000/SS)
/100);INT (P(J)*10000/SS)/100
4100 NEXT J
4105 PRINT AT 18,X;Y$
4107 IF UU=1 THEN GOTO 6020
4110 PRINT "TOT. GASTOS $A";AT 1
9,Y-LEN STR$ INT SS;SS;AT Y-P1,X
;Y$
4115 PRINT AT Y,P3;"1-MENU   2-C
OPY   3-CONT."
4120 GOSUB 1900
4122 IF A$="2" THEN COPY
4123 IF A$="3" THEN GOTO 4400
4124 IF A$="1" THEN GOTO 4130
4125 GOTO 4120
4130 CLS
4135 GOTO 95
4400 IF UU=1 THEN GOTO 4130
4402 IF M1=49 THEN GOTO 4450
4405 IF M1=33 THEN GOTO 4440
4410 IF M1=17 THEN GOTO 4430
4415 LET M1=17
4417 LET M2=32
4420 GOTO 4005
4430 LET M1=33
4432 LET M2=48
4434 GOTO 4420
4440 LET M1=49
4442 LET M2=60
4444 GOTO 4420
5000 PRINT AT P3,R;"■";AT Y,X;"P
RESIONE RECORD Y LUEGO ENTER"
5005 GOSUB 1900
5013 CLS
5015 SAVE "CUENTA"
5017 IF L3=P1 THEN GOTO 5100
5038 LET L2=X
5040 CLS
5045 GOTO 95
5100 LET L3=X
5105 FOR J=P2 TO C
5110 LET Z$(J, TO )=X$( TO Y)
5115 LET Z(J)=X
5120 NEXT J
5125 LET C=P1
5130 GOTO 5038
6000 CLS
6001 LET M1=61
6002 LET M2=70
6006 PRINT "ANALISIS DE INGRESOS
 AL";AT X,Y+P3;F$;"/";M$;"/";U$;
AT P1,X,Y$
6007 LET UU=1
6010 GOTO 4015
6030 PRINT "TOT. INGRESOS $A";AT 1
9,Y-LEN STR$ INT SE;SE;AT Y-P1,X
;Y$
6040 GOTO 4115
6500 PRINT B$(J, TO );"$A";AT LL
,Y-LEN STR$ INT P(J);P(J);AT LL,
28-LEN STR$ INT ((P(J)*10000/SE)
/100);INT (P(J)*10000/SE)/100
6510 GOTO 4100
9000 PRINT AT P3+P1,P1;"■";AT Y,
X;"1-COD.EGR. 2-COD.ING. 3-COD.D
EB."
9005 LET A$=INKEY$
9007 IF A$="0" THEN GOTO 2035
9010 IF A$<"1" OR A$>"3" THEN GO
TO 9005
9012 LET LLL=VAL A$
9015 CLS
9020 IF A$>"1" THEN GOTO 9025
9022 LET M1=P1
9023 LET M2=30
9024 GOTO 9032
9025 IF A$="3" THEN GOTO 9029
9026 LET M1=61
9027 LET M2=70
9028 GOTO 9032
9029 LET M1=71
9030 LET M2=ZX
9032 REM
9034 PRINT TAB 7;G$(LLL, TO
9035 GOSUB 9400
9037 PRINT AT Y,X;"1-MENU  2-A/B
/M  3-COPY  4-CONT."
9040 LET A$=INKEY$
9045 IF A$="2" THEN GOTO 9070
9050 IF A$="1" THEN GOTO 9060
9052 IF A$="3" THEN GOTO 9600
9053 IF A$="4" AND M2=30 THEN GO
TO 9700
9055 GOTO 9040
9060 CLS
9065 GOTO 95
9070 GOSUB 600
9075 PRINT AT Y,X;E$;"CODIGO"
9080 INPUT I
9085 IF I<M1 OR I>M2 THEN GOTO 9
080
9090 GOSUB 9500
9095 PRINT AT Y,X;E$;"DETALLE"
9100 PRINT AT X1,Y1;X$( TO R);AT
 X1,Y1;"?"
9101 INPUT A$
9102 IF A$="" THEN GOTO 9115
9103 IF LEN A$>R THEN GOTO 9101
9104 LET B$(I, TO )=A$
9110 GOSUB 600
9115 PRINT AT X1,Y1;B$(I, TO )
9120 PRINT AT Y,X;"OTRO CAMBIO?"
;R$
9125 LET A$=INKEY$
9130 IF A$="S" THEN GOTO 9070
9135 IF A$="N" THEN GOTO 9060
9140 GOTO 9125
9145 PRINT TAB P3+P3;"TABLA DE C
ODIGOS"
9150 GOTO 9035
9400 FAST
9401 PRINT Y$,
9402 FOR J=M1 TO M2-P1 STEP P2
9405 IF J<=P5 THEN GOTO 9440
9410 PRINT J;"-";B$(J, TO );"  "
;
9415 IF J+P1<=P5 THEN GOTO 9450
9420 PRINT J+P1;"-";B$(J+P1, TO
)
9425 NEXT J
9427 SLOW
9428 PRINT AT 19,X;Y$
9429 SLOW
9430 RETURN
9440 PRINT "0";
9445 GOTO 9410
9450 PRINT "0";
9455 GOTO 9420
9500 IF I>70 THEN LET II=I-70
9502 IF I>60 AND I<=70 THEN LET
II=I-60
9504 IF I>30 AND I<=60 THEN LET
II=I-30
9506 IF I>P1 AND I<=30 THEN LET
II=I
9508 IF I>=P1 AND I<=30 THEN LET
 II=I
9509 LET X1=INT (II/P2)
9510 IF II/P2-X1=X THEN GOTO 952
0
9511 LET Y1=P3
9512 LET X1=X1+P3
9515 RETURN
9520 LET Y1=Y-P2
9522 LET X1=X1+P2
9525 GOTO 9515
9600 COPY
9605 GOTO 9050
9700 CLS
9705 LET M1=31
9710 LET M2=60
9715 GOTO 9034
9800 CLS
9805 PRINT AT 3,0;"SALDO AL: ";F$
;"/";M$;"/";U$;"  $A ";INT (SAL1
*100)/100
9806 PRINT Y$
9810 GOSUB 3700
9815 GOTO 3852
9999 PRINT (Z$(J,P7 TO P8))
```




2do CONCURSO TRIMESTRAL



PATROCINA

SANWA

1er. Premio: Una Consola COMMODORE 64
2do. Premio: Una Consola SPECTRUM
3er. Premio: Dos Pasajes a Bariloche
4to. Premio: Un Grabador para Computadora
5to. Premio: Una Impresora Alpha Com 32

Condiciones para participar en el certamen:

1: Los programas deberán ser originales e inéditos, pudiendo cubrir todas las áreas: educativos y de cálculo, uso comercial, entretenimiento, personal y utilitarios para programación, hasta 64 K. 2: El criterio de elección se basará en: originalidad de la idea, método de programación, efectos gráficos y/o sonoros, documentación, presentación y ahorro de memoria. 3: Se enviarán a K-64 grabados en un cassette y acompañados por el listado correspondiente con pantallas y explicacion sobre la utilidad y manejo del programa. 4: Puede remitirse más de un programa por cassette, en lo posible grabados dos veces, para mayor seguridad. 5: El concurso K-64 se realizará trimestralmente. El cierre de la recepción de los trabajos será el 15/9/85. 6: K-64 se reserva el derecho de publicación de los programas enviados (como asimismo de la devolución del material recibido).
El cassette deberá ser enviado con su caja y con los datos del programa y del autor, como así también de la computadora para la cual está destinado.

SELECCION MENSUAL

Mensualmente se seleccionarán 50 Programas, los que se harán acreedores a los siguientes premios: Calculadoras, Máquinas de fotos, cassettes con programas, cassettes vírgenes, Becas para Cursos, etc.
Los Programas seleccionados continúan en Concurso para la gran final Trimestral.
RETIRO DE CUPONES: K-64: Cerrito 1320 - SANWA: Av. Corrientes 2198, Florida 683 - EPI: Suipacha 946 - 1er. piso, Viamonte 1479 - 8º Piso "B", Florida 683, Av. Corrientes 2198 - Radio del Plata: Av. Santa Fe 2043.

AUSPICIAN





ZX Spectrum / TK 90X / TS 2068

SCROLL INFINITO:

A veces resulta pesado, sobre todo durante un juego cuando la máquina solicita un "scroll?".

Este mensaje está controlado por una variable de sistema que cuenta el número de pantallas que se sucederán antes de que le pregunte scroll?.

La línea: 10 POKE 23692,255 deshabilitará ésto permitiéndole un descanso de hasta 254 pantallas sin tener que tocar el teclado.

CRASH DE LUJO:

El poner la Spectrum en "crash" se ha vuelto un deporte de los programadores desprevenidos.

Es interesante sin embargo provocarlo a propósito, deliberadamente, y una de las mejores maneras es con:

PRINT USR 1000

La máquina deberá dar un buzz, el teclado ingobernable y un borde negro aparece en pantalla.

Para demostrar ante los demás que uno domina los secretos del lenguaje de máquina de un modo elegante, trate esto:

PRINT "K 64 ES LO MAXIMO!"

y a continuación el PRINT USR.

No se puede borrar lo que queda en pantalla, y la única manera de pararlo es desenchufándola.

Para hacerlo más colorido,

RAND USR 5050

Aparece un rectángulo multicolorido, seguido por el logo de Sinclair, reseteando a la máquina.

Finalmente, si por casualidad tiene conectada una impresora Epson FX-80, trate:

RAND USR 600



CONVERSION DE PROGRAMAS DE DISTINTAS COMPUTADORAS

Aquel que se detenga a observar detenidamente los distintos programas que se publican en K64 habrá notado muchas diferencias entre los Basic's de cada marca y modelo de computadora.

A pesar de esto a veces es posible "traducir" los distintos dialectos entre sí y poder aprovechar un programa interesante.

Usualmente los más sencillos de traducir son aquéllos que tratan con fórmulas matemáticas o con almacenamiento y manipulación de datos. Los que por otro lado, incluyan gráficos, subrutinas en lenguaje de máquina o manejo de archivos externos presentarán seguramente dificultades insalvables.

Sin embargo, aquí van algunos consejos:

* Buscar en el programa que se desea convertir si tiene instrucciones del tipo PEEK, POKE, CALL, POP, USR, etc.

Son las que tienen que ver con el lenguaje de máquina. Para poder adaptarlas por lo general se necesitará conocer a fondo ambas computadoras, sus mapas de memoria, etc. Si no se trata de algo sencillo por lo general conviene reescribir todo el programa desde cero.

* Verificar ciertas instrucciones que tienen significados distintos en cada Basic, como: COLOR, DRAW, ERROR, GET, IF, THEN, PEEK, POINT, PRINT USING, PUT, SCREEN.

* Anótense todos los nombres de variables y lo que éstas representan.

* Divídase el programa en bloques funcionales o diagramas lógicos o de flujo.

* Las instrucciones de graficación y símbolos gráficos suelen ser muy particulares de cada modelo de computadora aunque a veces estudiando el efecto de esas, pueden lograrse resultados positivos.

* No conviene la traducción línea a línea, sino por bloques funcionales separados, aprovechando las capacidades especiales de la versión de Basic con la que se está trabajando.

CZ1000/1500 TK 83/85

CLEAR ANTES DE SAVE:
Si no es necesario SAVEar los valores de las variables junto con el programa, el comando CLEAR permite reducir la cantidad de información que enviará al cassette. Esto acorta a veces drásticamente el tiempo de grabado y luego el de reproducción, reduciendo además la probabilidad de error.

FUNCION VAL:
Si se usa frecuentemente una expresión dentro de un programa, conviene más definirla como un string, usando luego la función VAL para ejecutarla cuando sea necesario. Por ejemplo:
LET Z=SQR (X*X*Y*Y)
podríamos colocarla al principio del programa como:
LET Z$= "SQR (X*X*Y*Y)"
Usándola más tarde con:
LET Z=VAL Z$

CENTRANDO TITULOS:
Para que salga centrado en pantalla una palabra o un pedazo del texto almacenado en A$, por ejemplo:
PRINT TAB 16-LEN A$/2;A$
o
PRINT AT N,16-LEN A$/2;A$
Para justificar a la derecha:
PRINT TAB 32-LEN A$,A$
o
PRINT AT N,32-LEN A$;A$

POKE 16509:
Sirve para proteger nuestros programas de los piratas...
Primero, teclear las líneas a proteger.
Luego entrar POKE 16509,40
Esto cambiará el número de la primer línea por un número que comienza con una letra.
Luego entrar el resto del programa. Esas líneas estarán siempre presentes al final del listado y no podrán ser borradas o alteradas por lo menos de la manera habitual, y no hay un modo sencillo de como lograr otro POKE para volver a la situación inicial.

NUMEROS ALEATORIOS:
Si usa la función RND por sí sola, obtendrá números aleatorios que caen entre 0 y 1, con sus correspondientes decimales. Trate haciendo PRINT RND y vea el efecto. Para obtener un rango de números aleatorios enteros, hay que usar la función INT. Por ejemplo para simular el resultado de tirar un dado hay que obtener un número entre 1 y 6.
La fórmula general es:
INT(RND*máximo)+mínimo
que en el caso del dado sería:
PRINT INT(RND*6)+1
Si en cambio se quieren obtener enteros entre 50 y 100:
PRINT INT(RND+100)+50
Estos números aleatorios no se calculan, se sacan de tablas. Si por ejemplo uno hace RAND 3, la máquina comenzará dando el tercer número de la tabla y luego el cuarto, etc. Pero si ponemos RAND solamente, el primer número a obtener será muy difícil de preveer. Esto normalmente resulta bastante aleatorio para las aplicaciones comunes.

PROBANDO EL RAMTOP:
Si tiene un programa que requiere que la variable Ramtop sea corregida antes de cargar el programa, es útil agregarle un chequeo como el siguiente:

```
10 IF PEEK 16388+256*PEEK
16389=17388 THEN GOTO 20
11 PRINT "RAMTOP EQUIVOCA-
DO"
12 STOP
20 ETC...
```

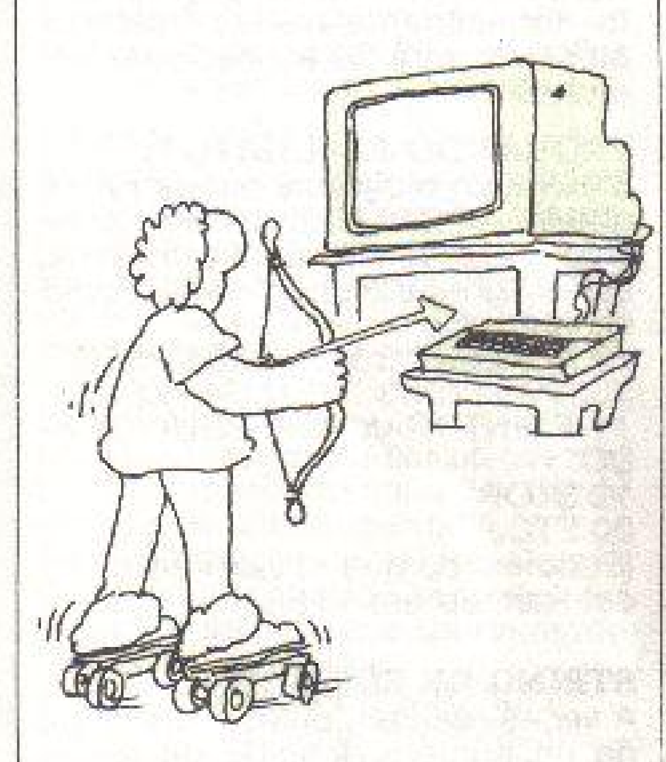
(Suponiendo que el valor correcto del Ramtop era 17388.

STRING DE ESPACIOS:
A veces resultan útiles los strings de un número definido de espacios. La mejor manera de lograrlo es con un DIM.
Por ejemplo:
DIM A$ (32)
genera un string A$ de 32 espacios.

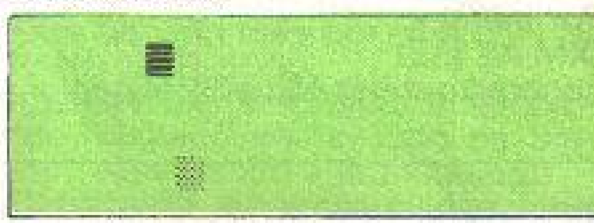
CUBITO

COMP: CZ1000/1500 TK83/85
CONF: 2 K
CLAS: ENT

Este juego de reflejos y rapidez consiste en atrapar, moviendo el punto negro con las flechas, al punto gris parpadeante, lo más rápido posible.



PANTALLA

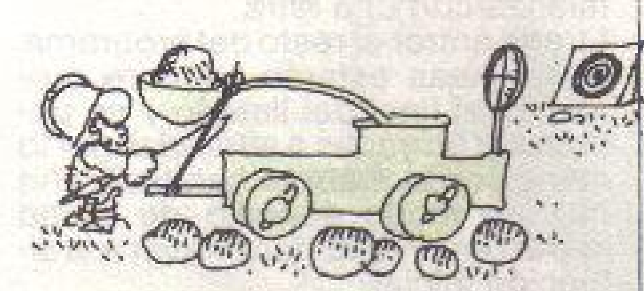


```
   1 REM *************
   2 REM *    K 64    *
   3 REM *************
   4 RAND
   5 PRINT
   6 PRINT "HAY QUE ATRAPAR AL C
UBITO EN EL MENOR TIEMPO POSIBLE
"
   7 PAUSE 200
   8 CLS
  10 LET C=50
  20 LET A=0
  30 LET B=0
  40 LET X=INT (10*RND)
  50 LET Y=INT (10*RND)
  60 PRINT AT X,Y;"▒"
  70 FOR N=1 TO C
  80 PRINT AT A,B;"■"
  90 PRINT AT A,B;" "
 100 IF INKEY$="5" THEN LET B=B-
1
 110 IF INKEY$="6" THEN LET A=A-
1
 120 IF INKEY$="7" THEN LET A=A+
1
 130 IF INKEY$="8" THEN LET B=B+
1
 140 IF ABS A=ABS X AND ABS B=AB
S Y THEN GOTO 180
 150 NEXT N
 160 PRINT "SUERTE DURA.. *SCORE
* ";100-C*2;" PUNTOS*"
 170 PAUSE 4E4.
 171 CLS
 172 GOTO 0
 180 PRINT "BIEN HECHO  *SCORE=
";C-N;"*"
 190 PAUSE 50
 200 LET C=C-1
 210 CLS
 220 GOTO 20
 250 SAVE "CUBITO
 260 RUN
```

CATAPULTA

COMP: CZ1000/1500 TK83/85
CONF: 2 K
CLAS: ENT

Hay que pegarle al cuadradito gris, direccionando adecuadamente la "piedra" en ángulo y velocidad.



PANTALLA



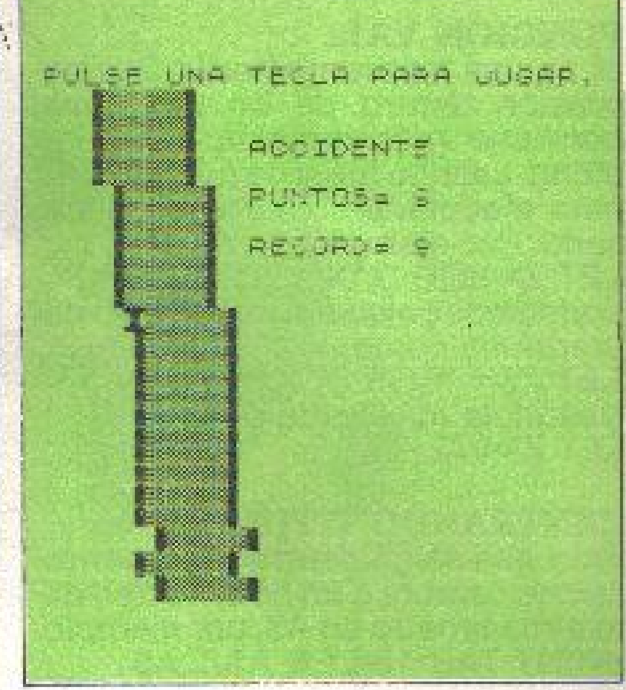

```
   1 REM *************
   3 REM *    K 64    *
   4 REM *************
   5 REM *CATAPULTA *
   6 REM *************
   7 REM
   8 RAND
   9 LET J=20
  10 LET A=28
  20 LET B=INT (RND*J)
  30 LET C=SGN PI
  40 LET D=RND/5+.5
  50 LET E=D
  65 CLS
  69 PRINT AT 0,4;"*CATAPULTA*"
  70 LET D=NOT PI
  80 PRINT AT B,A;"▒"
  90 PRINT AT D,D;"■";C,,,,"VELO
CIDAD? ";
 100 INPUT F
 110 PRINT F,,"ANGULO? ";
 120 INPUT G
 130 PRINT G
 140 LET G=G*.0174
 150 LET H=INT (F/A*D*SIN G-(4.9
*D*D))
 160 LET I=INT (D*F/A*COS G)
 170 IF I>PI*J OR H>J+J+PI OR H<
NOT PI THEN GOTO 260
 180 PLOT I,H
 190 LET D=D+E
 200 IF INT (I/2)=A AND J+SGN PI
-INT (H/2)=B THEN GOTO 220
 210 GOTO 150
 220 UNPLOT I,H
 230 PRINT "PUM"
 231 IF C<A/PI THEN PRINT "BRAVO
"
 233 IF C>A THEN PRINT "DEJA QUE
 DESEAR..."
 240 PAUSE A**PI
 250 GOTO SGN PI
 260 LET D=E
 270 PAUSE A**PI
 280 LET C=C+1
 290 GOTO PI*J
 300 SAVE "CATAPULT█"
 301 RUN
```

ACCIDENTE

COMP: SPECTRUM/CZ2000/TK90X
CONF: 16 K
CLAS: ENT

Conduce tu auto en una carretera tenebrosa evitando los accidentes. Se maneja con la "Z" y la "M".

PANTALLA




```
   1 PRINT AT 5,12;"ACCIDENTE":
INVERSE 0: PRINT AT 7,1;"CONDUZC
A SU AUTO              POR LA
CARRETERA "'" CONSIGUIENDO EL MA
YOR"'" NUMERO DE PUNTOS"'" MOVI
MIENTO DCHO E IZDO CON Z-M": PRI
NT AT 21,1; PAPER 4; INK 1;; FLA
SH 1;"PULSE UNA TECLA PARA JUGAR
.": PAUSE 0: CLS
   2 FOR t=0 TO 7: READ a,b
   3 POKE USR "p"+t,a: POKE USR
"o"+t,b
   4 NEXT t
   5 DATA 170,255,85,255,170,60,
85,60
   6 DATA 170,60,85,60,170,255,8
5,255
   7 LET c$="ACCIDENTE "
  10 LET H=0
  20 LET S=0
  25 LET X=5: LET A=3: CLS
  30 FOR N=0 TO 20
  40 PRINT TAB A; INK 1;"█"; INK
 2;"▒▒▒▒▒"; INK 1;"█"
  50 LET A=A+(A<7 AND A)*SIN (RN
D-.5)+(NOT A)-(A=7)
  60 NEXT N
  70 LET A=X
  80 FOR N=1 TO 20
  90 LET P=ATTR (N,X)
 110 PRINT AT N-1,A; INK 2;"▒";A
T N,X; INK 4;"I"
 120 IF P=57 THEN GOTO 170
 130 LET A=X
 140 LET X=X+(INKEY$="m")-(INKEY
$="z")
 145 FOR j=1 TO 30: NEXT j
 150 NEXT n
 160 LET s=s+n: GOTO 25
 170 LET s=s+n
 175 FOR a=1 TO 10: BORDER RND*7
: SOUND .1,a-10
 180 PRINT AT 2,10+a; INK RND*5;
c$(a)
 183 NEXT a
 185 PRINT AT 4,11;"PUNTOS= ";s
 190 IF h<s THEN LET h=s
 200 PRINT AT 6,11; INK 1; FLASH
 1;"RECORD= ";H
 210 INPUT "OTRA PARTIDA ?";A$
 220 IF A$(1)="s" THEN GOTO 20
 230 IF A$(1)="n" THEN GOTO 1000
1000 STOP
1070 STOP
9999 SAVE "CARRERA" LINE 1
```

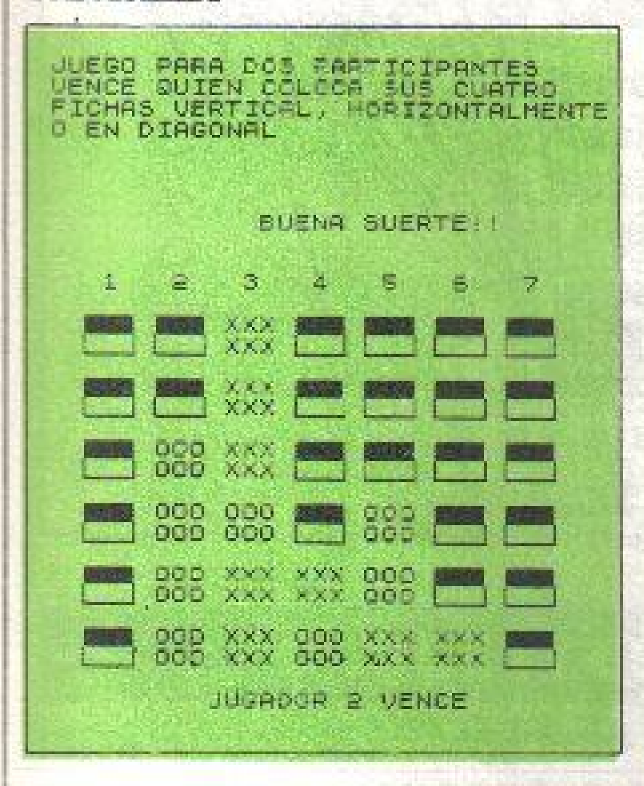


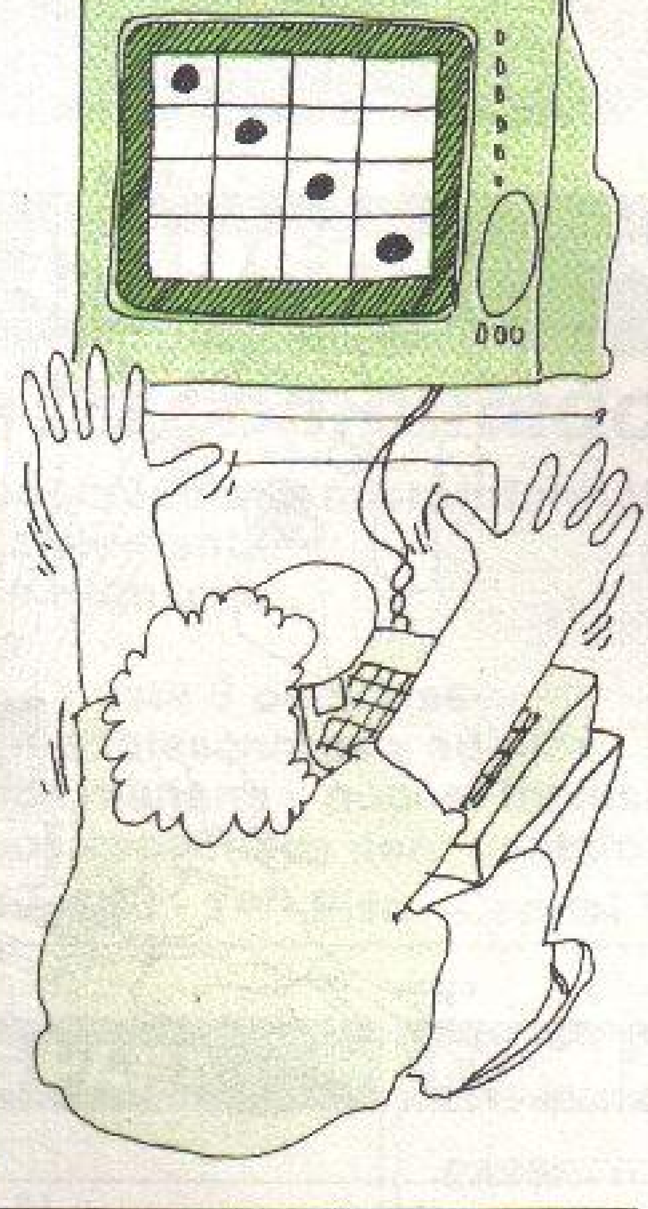

TA TE TI 4

COMP: SPECTRUM/CZ2000/TK90X
CONF: 16 K
CLAS: ENT

El clásico TA-TE-TI, pero un poco más difícil. Gana el primero que ubica las cuatro fichas en forma horizontal, vertical o en diagonal.

PANTALLA





```
   1 CLS
   5 BORDER 6: PAPER 7: INK 0
  10 PRINT AT 7,0;"JUEGO PARA DOS PARTICIPANTES": PRINT AT 8,0;"VENCE QUIEN COLOCA SUS CUATRO": PRINT AT 9,0;"FICHAS VERTICAL, HORIZONTALMENTE": PRINT AT 10,0;"O EN DIAGONAL": PRINT AT 17,12;"BUENA SUERTE!!"
  20 PAUSE 250: PRINT AT 21,2;"pulse una tecla para jugar"
  30 PAUSE 0: CLS
 100 LET X=1: LET I$="XXX": INK 0: PAPER 7: CLS : DIM A(12,13)
 110 LET S$=CHR$ 144+CHR$ 145+CHR$ 146
 120 LET T$=CHR$ 147+CHR$ 95+CHR$ 148
 130 LET U$="                "
 140 DATA 0,127,127,127,127,127,127,127
 150 DATA 0,255,255,255,255,255,255,255
 160 DATA 0,254,254,254,254,254,254,254,
 170 DATA 128,128,128,128,128,128,128,255
 180 DATA 1,1,1,1,1,1,1,255
 190 FOR J=144 TO 148: FOR K=0 TO 7
 200 READ A: POKE USR CHR$ J+K,A
 210 NEXT K: NEXT J
 220 FOR J=2 TO 17 STEP 3
 230 FOR K=2 TO 26 STEP 4
 240 PRINT INK 6; PAPER 0;AT J,K;S$
 250 PRINT PAPER 6;AT J+1,K;T$
 260 NEXT K: NEXT J
 270 PRINT INK 3;AT 0,3;"1   2   3   4   5   6   7"
 280 PRINT INK 7; PAPER X;AT 20,11;"JUGADOR ";X
 290 INPUT "QUE COLUMNA [ 1 A 7 ] ";A
 300 IF A<>INT A OR A>7 OR A<=0 THEN GOTO 290
 310 LET C=(A-1)*4+2
 320 IF ATTR (2,C)<>6 THEN PRINT FLASH 1;AT 21,7;"COLUMNA ";A;" ESTA COMPLETA": GOTO 290
 330 PRINT AT 21,7;U$
 340 FOR J=0 TO 18
 350 LET Z=ATTR (J,C)
 360 PRINT AT J,C; PAPER X; INK 7;I$
 370 IF J=2 THEN PRINT INK 3;AT J-2,C;CHR$ 32,A,CHR$ 32
 380 IF J>2 AND Z=6 THEN PRINT PAPER 6;AT J-2,C;T$
 390 IF J>2 AND Z=48 THEN PRINT AT J-2,C;"   "
 400 IF J>2 AND Z=56 THEN PRINT INK 6; PAPER 0;AT J-2,C;S$
 410 IF J=18 OR ATTR (J+2,C)=23 OR ATTR (J+2,C)=15 THEN SOUND 2.1,-20: GOTO 440
 420 NEXT J
 440 LET L=J/3+3: LET C=A+3
 450 LET A(L,C)=X
 460 FOR J=-3 TO 0
 470 IF A(L+J,C)=X AND A(L+J+1,C)=X AND A(L+J+2,C)=X AND A(L+J+3,C)=X THEN GOTO 550
 480 IF A(L,C+J)=X AND A(L,C+J+1)=X AND A(L,C+J+2)=X AND A(L,C+J+3)=X THEN GOTO 550
 490 IF A(L+J,C+J)=X AND A(L+J+1,C+J+1)=X AND A(L+J+2,C+J+2)=X AND A(L+J+3,C+J+3)=X THEN GOTO 550
 500 IF A(L-J,C+J)=X AND A((L-J)-1,C+J+1)=X AND A((L-J)-2,C+J+2)=X AND A((L-J)-3,C+J+3)=X THEN GOTO 550
 510 NEXT J
 520 LET I$="OOO": IF X=2 THEN LET I$="XXX"
 530 LET X=X+1: IF X=3 THEN LET X=1
 540 GOTO 280
 550 PRINT OVER 1; FLASH 1;AT 20,11;U$( TO 8)
 560 PRINT AT 20,9; FLASH 1; INK X;"JUGADOR ";(X);" VENCE": FOR a=1 TO 5: FOR b=20 TO 50 STEP 5: SOUND .01,b: SOUND .1,b-5: NEXT b: NEXT a
 570 INPUT "OTRA PARTIDA ";A$
 571 IF A$="" THEN GOTO 571
 572 IF A$="s" THEN RUN
 573 IF A$="n" THEN GOTO 700
 580 PRINT AT 20,9;"              ": GOTO 220
 590 PRINT AT 21,0;U$;AT 21,16;U$
 600 DIM a(12,13): GOTO 220
 700 STOP
 765 STOP
9999 SAVE "TA TE TI 4" LINE 1
```

EL MAESTRO

COMP: CZ1000/1500 TX83/85
CONF: 2 K
CLAS: EDU

Este programa está preparado para los más chicos.
Consiste en resolver 10 operaciones matemáticas elementales, de suma, resta, multiplicación y división.
Hay 5 niveles de dificultad, y al final de los ejercicios la computadora realiza la evaluación, felicitando o aconsejando.

PANTALLA

```
MI NOMBRE ES SINCLAIR
Y EL TUYO ?  JUAN
ELIGE LA OPERACION
+ 1  - 2  * 3  / 4
ELIGE EL NIVEL: DE 1 A 5
```

```
   1 REM ***   K 64   ***
   2 REM
   3 RAND
   4 LET F=0
   5 LET A$="+-*/"
   6 CLS
   9 PRINT "MI NOMBRE ES SINCLAIR";AT 2,0;"Y EL TUYO ? "
  10 INPUT N$
  11 PRINT N$
  13 PRINT
  15 PRINT "ELIGE LA OPERACION"
  16 PRINT
  17 PRINT "+ 1  - 2  * 3  / 4"
  20 INPUT A
  30 PRINT ,,"ELIGE EL NIVEL; DE 1 A 5"
  40 INPUT B
  50 FOR N=1 TO 10
  60 CLS
  65 PRINT "OPERACION ";N;" CORRECTAS ";F
  70 LET C=INT (10**B*RND)
  80 LET D=INT (10**B*RND)
  90 IF A>0 THEN LET D=INT (D/(10**(B-1)))+1
 100 LET B$=STR$ C+" "+A$(A)+" "+STR$ D
 110 PRINT ,,B$;" = ";
 120 INPUT D
 125 PRINT D
 130 IF ABS (VAL B$-D)>.01 THEN GOTO 170
 140 PRINT ,,"CORRECTO,MUY BIEN ";N$;AT 20,0;"AHORA, ENTER"
 160 LET F=F+1
 165 GOTO 180
 170 PRINT ,,"MAL",,,,"EL RESULTADO CORRECTO ES ";VAL B$;AT 20,0;"AHORA, ENTER"
 180 INPUT D$
 185 CLS
 190 NEXT N
 195 PRINT ,,N$
 200 PRINT "ACERTASTE ";F
 201 PRINT "DE 10 OPERACIONES"
 203 PRINT ,,("REGULAR , PRACTICA MAS PARA    MEJORAR" AND (F<6))+("BIEN, PUEDES MEJORAR UN POCO MAS" AND (F>5 AND F<8))+("MUY BIEN , CONTINUA ASI" AND (F>7 AND F<10))+("EXCELENTE . OBTUVISTE UN OPTIMO RESULTADO . CONTINUA ASI." AND (F=10))
 208 PRINT AT 20,0;"QUIERES CONTINUAR (S/N)"
 210 INPUT D$
 215 CLS
 220 IF D$="S" THEN GOTO 3
 240 SAVE "MAESTRO"
 250 GOTO 1
```



ORGANITO

COMP: CZ1000/1500 TK83/85
CONF: 1 K
CLAS: ENT

Este complicadísimo y extenso programa convierte a tu computadora en un perfecto micro órgano, aunque bastante rudimentario...
Se usan las teclas 1 a 9 para tocar. Por supuesto que para escuchar, hay que conectar un pequeño amplificador en la salida de MIC de la computadora aunque algunos grabadores permiten monitorear lo que se está grabando conectándole un audífono en EAR y poniéndolos a grabar con PLAY/RECORD.

```
  5 FAST
 10 GOTO CODE INKEY$
 28 RAND
 29 RAND
 30 RAND
 31 RAND
 32 RAND
 33 RAND
 34 RAND
 35 RAND
 36 RAND
 37 RAND
200 SLOW
210 RUN
```



ECUACIONES

COMP: CZ1000/1500 TK83/85
CONF: 16 K
CLAS: EDU

La resolución de ecuaciones del tipo N x N resulta engorrosa hacerla a mano.

Este corto programa facilita las cosas para nosotros. Es fácil de usar, las instrucciones salen por pantalla.

PANTALLA

```
ECUACION 1              6 ECUACIONES
 X1:  100
 X2:  20
 X3:  40
 X4:  12
 X5:  1000
 X6:  3
```

```
   1 REM
   2 REM +
   3 REM ************+
   4 REM *ECUACIONES*
   5 REM ************+
   6 REM
  10 DIM O$(32)
  15 LET J=1
  20 LET L=0
  30 LET I=0
  40 PRINT AT 21,0;"CUANTAS VARI
ABLES TIENE UD.?"
  50 INPUT N
  60 PRINT AT 21,0;O$
  70 DIM Y(N)
  80 DIM X(N)
  90 DIM D(N+1,N+1)
 100 DIM C(N+1,N+1)
 110 FOR A=1 TO N
 120 PRINT AT 0,18;N;" ECUACIONE
S"
 130 PRINT AT 0,0;"ECUACION ";A
 140 PRINT AT 18,0;"PARA CADA EC
UACION ENTRE CADA   COEF.(INCLUY
ENDO EL TERMINO IN--DEPENDIENTE)
,LUEGO ENTER"
 150 FOR B=1 TO N+1
 160 PRINT AT B,0;" X";B
 170 INPUT D
 180 LET C(A,B)=D
 190 LET D(A,B)=D
 200 PRINT ": ";C(A,B)
 210 NEXT B
 220 PRINT AT 18,0;O$;"PARA CORR
ECCIONES:ESCRIBIR ""C"" ENTER,N
UEVO COEFICIENTE ."
 230 PRINT AT 21,0;"ESCRIBIR ENT
ER PARA CONTINUAR"
 240 INPUT E$
 250 CLS
 260 IF E$="C" THEN GOTO 120
 270 CLS
 280 NEXT A
 300 REM PIVOT Y CALCULO
 310 FOR A=1 TO N
 320 LET P=C(A,A)
 330 FOR B=A+1 TO N
 340 IF ABS P>=ABS C(B,A) THEN G
OTO 410
 350 LET P=C(B,A)
 360 FOR D=1 TO N+1
 370 LET F=C(A,D)
 380 LET C(A,D)=C(B,D)
 390 LET C(B,D)=F
 400 NEXT D
 410 NEXT B
 420 FOR D=N+1 TO A STEP -1
 430 IF C(A,A)=0 THEN GOTO 800
 440 LET C(A,D)=C(A,D)/C(A,A)
 450 NEXT D
 460 FOR B=A+1 TO N
 470 FOR D=N+1 TO A STEP -1
 480 LET C(B,D)=C(B,D)-C(A,D)*C(
B,A)
 490 NEXT D
 500 NEXT B
 510 NEXT A
 550 REM CALCULO
 560 LET X(N)=C(N,N+1)
 570 FOR A=N-1 TO 1 STEP -1
 580 LET X(A)=C(A,N+1)
 590 FOR D=N TO A+1 STEP -1
 600 LET X(A)=X(A)-C(A,D)*X(D)
 610 NEXT D
 620 NEXT A
 650 REM SALIDA
 660 SLOW
 670 FOR A=1 TO N
 680 FOR B=1 TO N-1
 690 PRINT D(A,B);"*X";B;"+ ";
 700 NEXT B
 710 PRINT D(A,N);"*X";N;" = ";D
(A,N+1) TAB 0
 715 PAUSE 700
 716 CLS
 720 NEXT A
 730 IF L=1 THEN RETURN
 740 FOR A=1 TO N
 750 PRINT AT 20-N+A,2;"X";A;" =
",TAB 4+(1 AND X(A)>=0);X(A)
 750 NEXT A
 770 GOTO 1000
 800 REM DEPENDIENTE O INCONSIST
ENTE
 810 FOR K=A TO N
 820 FOR H=A TO N
 830 LET C(K,H)=C(K,H+1)
 840 NEXT H
 850 LET C(K,N)=0
 860 NEXT K
 870 IF I=A THEN GOTO 910
 880 LET I=A
 890 LET J=I
 900 GOTO 320
 910 LET J=J+1
 920 IF J<N-A+1 THEN GOTO 320
 930 LET L=1
 940 GOSUB 360
 950 FOR K=1 TO N
 960 IF C(K,N+1)<>0 THEN PRINT A
T 21,0;"INCONSISTENTE"
 970 IF C(K,N+1)<>0 THEN GOTO 10
00
 980 NEXT K
 990 PRINT AT 21,0;"DEPENDIENTE"
1000 PAUSE 4E4
1001 CLS
1002 GOTO 1020
1010 SAVE "ECUACIONES"
1020 RUN
```

Los servicios de Epi

CURSOS:

Sólo Epi le da un computador para Ud. solo

- Introducción a la microinformática
- Basic elemental
- Basic avanzado
- Logo
- Grupo hasta 8 personas
- Niños adolescentes y adultos
- Turnos mañana y noche, inclusive sábados.
- Cursos especiales para colegios

Suipacha 946 - 1er. Piso - Capital TE.: 311-8618

CASSETTES PARA

TI-99/4A

MICRODIGITAL

COMMODORE 64

COMPILADOR para TI-99/4A

FORMATOS DISPONIBLES

CASSETTECAS

x 4: $a 6.250.-
x 6: $a 8.200.-
x 12: $a 12.500.-

SISTEMA SKINPACK

118 Títulos
a $a 1.600.- c/u

SISTEMAS

Las microcomputadoras son equipos aptos para procesar sistemas comerciales, verifíquelo!, EPI se lo asegura.

Disponemos:

- Stock
- Facturación
- Clientes
- Cuentas Corrientes
- Contabilidad
- Listas de precios

... y también sistemas a su medida.



EMPRESA PARA INFORMATICA

INSTITUTO: Suipacha 946 1er. Piso (1008) Capital.
VENTAS: Viamonte 1479 8° "B" (1055) Capital.
Teléfonos: 311-8618 y 49-7985.
Florida 683
Av. Corrientes 2198

SE ACEPTAN TARJETAS DE CREDITO

AHORRO DE MEMORIA DE SU 1000/1500

(Parte II)

¿CUANDO CONSUMIMOS 16 BYTES?:

10 LET A=SIN 1 SIN(seno)
El consumo es el mismo para:
COS (coseno)
ACOS (arco coseno)
INT (entero)
ABS (absoluto)
LN (logaritmo natural)
TAN (tangente)
ATAN (arco tangente)
SGN (signo)
SQR (raíz cuadrada)
EXP (función exponencial)

Utilizando el nombre de la variable en vez de números, volvemos a tener como antes una reducción del número de bytes.

10 LET A=SIN A en este caso el consumo es de.10 bytes.
10 DIM A (1) consume 16 bytes, pero
10 DIM A (10) consume 17 bytes
10 DIM A (100) consume 18 bytes.
10 DIM A (1,1) consume 24 bytes, de modo que " ,1" consume 8 bytes, pero presten atención nuevamente cuando colocamos letras en lugar de números
10 DIM A (B) consume 10 bytes (ahorro 6 bytes)
10 DIM A (B,C) consume 12 bytes (ahorro 12 bytes)

Cuando se escriben programas con variables con subíndice, el contador de línea no tomará en cuenta la reserva de lugar en la variable almacenada en memoria RAM para los elementos de un vector o de una matriz: concede 5 bytes por número para el caso de una variable unidimensional o vector.

Cuando tenemos variables bidimensionales o matrices, se multiplican los números encerrados entre paréntesis, y al resultado se lo multiplica por 5, entonces el valor resultante es el número de bytes que reserva la variable.

10 DIM A (5,10) reserva 250 bytes para la matriz a
10 PRINT PEEK (1) consume 16 bytes, pero
10 PRINT PEEK (10) consume 16 bytes, y
10 PRINT PEEK (100) consume 18 bytes.

El tamaño más usual que se encuentra en un PEEK es:
10 PRINT PEEK (10000) y el consumo es de 20 bytes.

¿CUANDO CONSUMIMOS 17 BYTES?:

10 DIM A$ (1) consume 1 byte más que 10 DIM A (1) de modo que todas las cadenas con subíndice consumirán un byte más que las correspondientes a variables numéricas.

Hay una importante diferencia entre los dos tipos de variables, y las alfanuméricas solamente ocupan un byte por caracter y no cinco, de manera tal que:
10 DIM A$ (5,10) reserva solamente 50 bytes.

¿CUANDO CONSUMIMOS 18 BYTES?:

10 PRINT INT (RND * 9)
El uso de la función RND se vio cuando tratamos el

En esta segunda parte, seguimos ofreciendo al lector los distintos consumos de memoria de esta microcomputadora. Esperamos que los tengan en cuenta a la hora de programar.

consumo de 7 bytes, pero la manera más usual de utilizarlo es ésta.
Nuevamente el ahorro que tenemos si empleamos 10 PRINT INT (RND X) es de 6 bytes, ya que en este caso consumimos 12 bytes.

¿CUANDO CONSUMIMOS 19 BYTES?

Las subrutinas consumen un mínimo de 19 bytes
1 GOTO 9
9 GOSUB 5
5 RETURN
en este caso 13 bytes toma el GOSUB y 6 el RETURN.

consume 32 bytes de modo que la subrutina insume 19 bytes.

¿CUANDO CONSUMIMOS 21 BYTES?:

10 POKE 1,1 pero
10 POKE 10,1 consume 22 bytes,
10 POKE 10000,1 consume 25 bytes
10 POKE 10000,10 consume 26 bytes
10 POKE A,A consume 9 bytes
10 PLOT 1,1 consume 21 bytes; pero 10 PLOT 1,10 consume 22 bytes
10 PLOT A,A consume 9 bytes
El comando UNPLOT tiene el mismo consumo.

¿CUANDO CONSUMIMOS 23 BYTES?:

10 LET A = 2 * 2 consume 23 bytes, pero
10 LET A = 2 consume 15 bytes, de modo que el consumo de un número y el signo de multiplicación es de 8 bytes.
Elevar un número al cuadrado consume la misma cantidad de memoria
10 LET A=2 ** 2 insume 23 bytes
10 LET A=−2 * −2 consume 25 bytes, y
10 LET A=− ** 2 consume 24 bytes, pero da error.
Todos los números al cuadrado son positivos y nuestra microcomputadora da una respuesta negativa, ésto significa que hay un error en el lenguaje.

Si tienen que escribir programas donde se deben elevar números al cuadrado, será necesario incluir una función ABS (absoluto) en todas las líneas de programa donde aparece "**", como por ejemplo:
10 LET A = (ABS −2)**2
10 FOR J = 1 TO 9 consume 23 bytes, pero
10 FOR J = 1 TO 10 consume 24 bytes, y
10 FOR J = 1 TO 9 STEP 2 consume 31 bytes, de modo que STEP 2 consume 8 bytes.
El consumo básico de un ciclo FOR/NEXT es:
10 FOR J = 1 TO 9   30 bytes.
20 NEXT J

Continuaremos con este tema, en el próximo número.

Ing. Julio José PUTRUELE
Ing. Miguel Angel MAUBRO

COMO CARGAR PROGRAMAS EN CODIGO MAQUINA USANDO CODIFICACION HEXADECIMAL

Este programa auxiliar escrito en Basic, permite cargar programas en código máquina, usando directamente los pares de números hexadecimales correspondientes, y ubicarlos dentro de una sentencia REM.

El programa en código máquina ya ingresado y verificado se puede grabar en cassette luego de eliminar las sentencias Basic auxiliares. En la sentencia REM de la primera línea se deben teclear tantos caracteres como Bytes tenga el programa a ingresar (en este ejemplo 11 caracteres).

Primero ejecutar el programa Basic con lo que se consigue "meter" el programa en Código Máquina que hay en el DATA dentro de los 100 Bytes que se blanquearon a partir de la dirección 32500 inclusive. Para ejecutar el programa en código máquina se deberá teclear PRINTSUR 32500 y en la última línea de la pantalla deberá aparecer para este ejemplo el número 43 (lo que hace este programa dado como ejemplo es colocar el nº 42 en un registro y sumarle un 1).

Una vez comprobado que el programa en código máquina funciona como se esperaba, se borran todas las líneas de sentencias Basic exceptuando la línea 10 REM (que es donde está guardado el programa en forma de símbolos aparentemente desordenados). Una vez que el listado del programa BASIC esté formado sólo por la sentencia REM se procederá a grabar dicho programa.

Autores: Ing. Julio José Putruele
Ing. Miguel Angel Maubro



```
 1Ø REM zzzzzzzzzzz
 15 CLEAR 32499
 18 LET PROG= PEEK 23635+ 256 * PEEK 23636
 2Ø LET X= PROG+ 5
 3Ø LET A$=" "
 35 IF A$=" " THEN READ A$
 4Ø IF A$="S" THEN STOP
 5Ø LET C= CODE A$
 55 LET D= CODE A$(2)
 6Ø IF C>=65 THEN LET C= C- 7
 7Ø IF D>=65 THEN LET D= D- 7
 8Ø POKE X , 16*C+D- 816
 9Ø LET X= X+ 1
1ØØ LET A$=A$( 3 TO)
11Ø GO TO 35
12Ø DATA "Ø6", "ØØ", "ØE", "2A", "21",
   "Ø1", "ØØ", "Ø9","44", "4D", "C9"
```

C U R S O S

BATALLA TIERRA-AIRE

TI 99/4A

```
8
4 CALL CLEAR
5 CALL SCREEN(12)
6 CALL COLOR(9,2,12)
7 CALL CHAR(96,"00000000F0F0F0F0")
8 CALL CHAR(97,"F0F0F0F0")
9 CALL CHAR(98,"FFFFFFFFF0F0F0F0")
10 CALL CHAR(99,"F0F0F0F0F0F0F0F0")
11 CALL CHAR(100,"FFFFFFFFFFFFFFFF")
12 CALL CHAR(101,"FF00FF00FF00FF00")
13 CALL CHAR(102,"F000F000F000F000")
14 CALL VCHAR(4,3,100,2)
15 CALL VCHAR(4,6,100,2)
16 CALL VCHAR(4,9,100,2)
17 CALL HCHAR(5,4,96)
18 CALL HCHAR(4,4,97)
19 CALL HCHAR(3,3,98)
20 CALL HCHAR(3,4,97)
21 CALL HCHAR(5,5,96)
22 CALL HCHAR(3,6,98)
23 CALL HCHAR(3,7,99)
24 CALL HCHAR(4,7,97)
25 CALL HCHAR(5,8,96)
26 CALL HCHAR(3,9,99)
27 CALL HCHAR(5,11,96)
28 CALL HCHAR(2,9,97)
29 CALL HCHAR(24,1,101,30)
30 CALL VCHAR(1,30,102,24)
31 LABEL$="EMPRESA PARA INFORMATICA"
32 F=8
33 C=3
34 GOSUB 58
35 LABEL$="PROGRAMAS PARA EL"
36 F=12
37 C=3
38 GOSUB 58
39 LABEL$="MICROCOMPUTADOR TI-99/4A"
40 F=14
41 C=3
42 GOSUB 58
43 LABEL$="FABRICADO Y DISTRIBUIDO  EN"
44 F=18
45 C=3
46 GOSUB 58
47 LABEL$="LA REPUBLICA  ARGENTINA POR"
48 F=20
49 C=3
50 GOSUB 58
51 LABEL$="'SDT'INTELIGENCIA ARGENTINA"
52 F=22
53 C=3
54 GOSUB 58
55 FOR TT=1 TO 500
56 NEXT TT
57 GOTO 64
58 FOR T=1 TO LEN(LABEL$)
59 A$=SEG$(LABEL$,T,1)
60 B=ASC(A$)
61 CALL HCHAR(F,C-1+T,B)
62 NEXT T
63 RETURN
64 CALL COLOR(9,2,8)
100 REM
170 CALL CLEAR
180 CALL SCREEN(14)
200 PRINT TAB(3);"INGRESE I PARA INSTRUCCIONES"
210 PRINT TAB(8);"N PARA SEGUIR ": : : : : : : : : : :
220 CALL KEY(0,LP,PL)
230 IF PL=0 THEN 220
240 IF LP=78 THEN 680
250 PRINT TAB(3);"* BATALLA TIERRA-AIRE *"
260 PRINT
270 PRINT "EL OBJETIVO DEL JUEGO ES EL"
280 PRINT "DE DESTRUIR A TODOS LOS AVIO"
290 PRINT "NES QUE PUEDA. PARA DISPARAR"
300 PRINT "SU MISIL OPRIMA Q. PARA IRSE"
310 PRINT "A LA IZQUIERDA S.CON D PARA "
311 PRINT "IR HACIA LA DERECHA."
330 PRINT
340 PRINT TAB(10);"* AVIONES *"
350 PRINT
360 PRINT "LOS AVIONES DISPARAN UN NUE-"
370 PRINT "VO TIPO DE LASER. LAS NAVES"
380 PRINT "PUEDEN VENIR DE AMBOS LADOS"
390 PRINT "DE LA PANTALLA."
430 PRINT : : :
440 PRINT "OPRIMA UNA TECLA PARA SEGUIR"
450 PRINT
460 CALL KEY(0,KL,LK)
470 IF LK=0 THEN 460
480 PRINT TAB(13);"* DEFENSA EN TIERRA *"
490 PRINT
500 PRINT "EL LASER NO PUEDE PENETRARLO"
510 PRINT "PERO LA BARRERA SOLO PUEDE "
520 PRINT "RESISTIR 5 IMPACTOS DIRECTOS"
530 PRINT "UD. PUEDE DISPARAR DE ATRAS"
540 PRINT "DE LA BARRERA, PERO ESO LA "
550 PRINT "DEBILITARA ANTE LOS ATAQUES".
555 PRINT "DIRECTOS DEL LASER ENEMIGO."
560 PRINT
570 PRINT TAB(10);"* PUNTAJE *"
580 PRINT
590 PRINT "SE ANOTARA DE ACURDO AL SI-"
600 PRINT "GUIENTE ACUERDO:20 PUNTOS SIDERRIBA
UN AVION QUE VUELA  ALTO. Y 5 PUN
TOS PARA AQUE-"
610 PRINT "LLOS QUE VUELAN BAJO."
620 PRINT "SI SU PUNTAJE SUPERA LOS 100PUNTOS,
HABRA GANADO UN JUEGOEXTRA."
640 PRINT : :
650 PRINT "OPRIMA UNA TECLA PARA SEGUIR"
660 CALL KEY(0,KL,LK)
670 IF LK=0 THEN 660
671 CALL CLEAR
672 CALL SCREEN(14)
673 PRINT "ADEMAS UD. VERA EN LA PANTA-LLA
DOS MARCADORES.EL PRIME-RO LE INDICA
EL PUNTAJE QUE "
674 PRINT "VA LOGRANDO. Y EL SEGUNDO ESLA
CANTIDAD DE IMPACTOS DI- RECTOS SOBRE
LA BARRERA."
675 PRINT : : : : : : : : :
676 PRINT "OPRIMA UNA TECLA PARA SEGUIR"
677 CALL KEY(0,KL,LK)
678 IF LK=0 THEN 677
680 CALL CLEAR
685 CALL SCREEN(14)
690 CALL COLOR(2,2,8)
700 PRINT "INGRESE NIVEL DE JUEGO :"
710 PRINT
720 PRINT " 1  PROFESIONAL"
730 PRINT
740 PRINT " 2  INTERMEDIO"
750 PRINT
760 PRINT " 3  NOVICIO"
770 PRINT
780 PRINT " 4  INICIADO"
781 PRINT
782 PRINT " 5  FIN DEL JUEGO"
790 PRINT : : : :
800 CALL KEY(0,DIF,XS)
801 IF DIF=53 THEN 805
802 GOTO 810
805 END
810 IF (DIF<49)+(DIF>52)=-1 THEN 800
820 CALL SCREEN(8)
830 DIFF=DIF-38
840 M$="0"
850 AA=1
860 ZZ=3
870 CALL CLEAR
880 REM
890 REM   BA$=BARRERA, L$=  LASER, R$=COMETE,
P$ & PR$= AVION, TL$ & TR$=TANQ, D
$=  DESTRUCCION
900 BA$="FFFFFFFF"
910 D$="2A0441228904S220"
920 L$="1818181818181818"
930 R$="181818181818 3C7E"
940 P$="603098FFFF983060"
950 PR$="060C19FFFF190C06"
960 TL$="071F7FFFFF7F3F0F"
970 TR$="E0F8FEFFFFFEFCE0"
980 CALL CHAR(120,D$)
990 CALL CHAR(144,L$)
1000 CALL CHAR(97,R$)
1010 CALL CHAR(112,TL$)
1020 CALL CHAR(113,TR$)
1030 CALL CHAR(101,BA$)
1040 CALL CLEAR
1050 BA=0
1060 CALL COLOR(2,16,2)
1070 CALL COLOR(16,2,2)
1080 CALL COLOR(15,12,8)
1090 CALL COLOR(13,3,3)
1100 CALL COLOR(12,7,8)
1110 T=15
1120 REM
1130 CALL HCHAR(24,T,112)
1140 CALL HCHAR(24,T+1,42)
1150 CALL HCHAR(24,T+2,113)
1160 CALL HCHAR(23,T+1,152)
1170 CALL HCHAR(22,15,101,3)
1180 REM BASE VERDE
1190 CALL HCHAR(23,1,128,15)
1200 CALL HCHAR(23,17,128,15)
1210 CALL HCHAR(24,1,128,14)
1220 CALL HCHAR(24,18,128,14)
1230 REM A=ALTURA DEL AVION
1240 RANDOMIZE
1250 A=2*INT(9*RND)+1
1260 IF A=1 THEN 1240
1270 REM DIRECCION AVION
1280 RANDOMIZE
1290 B=INT(2*RND)+1
1300 ON B GOTO 1310,1350
1310 B=30
1320 CALL CHAR(96,PR$)
1330 DIR=-2
1340 GOTO 1380
1350 DIR=2
1360 CALL CHAR(96,P$)
1370 B=2
1380 Y=21
1390 REM EMPIEZA
1400 CALL KEY(0,K,S)
1410 REM 81 DISP. COMETE,     83 Y 68 MUEVE TANQUE
1420 IF K=81 THEN 1450
1430 IF K=83 THEN 2640
1440 IF K=68 THEN 2770 ELSE 1550
1450 IF Y=21 THEN 1480 ELSE 1470
1460 REM DISP. COMETE
1470 CALL VCHAR(Y+2,T+1,32)
1480 CALL VCHAR(Y,T+1,97)
1490 TT=T+1
1500 Y=Y-2
1510 IF Y=-1 THEN 1520 ELSE 1550
1520 CALL VCHAR(1,T+1,32)
1530 GOTO 1380
1540 REM MUEVE AVION
1550 IF DIR=-2 THEN 1570
1560 IF B<3 THEN 1590
1570 CALL VCHAR(A,B-DIR,32)
1580 IF B<2 THEN 1900
1590 CALL VCHAR(A,B,96)
1600 RANDOMIZE
1610 Q=INT(15*RND)+1
1620 REM IMPACTO LASER AVION
1630 TT=TT-2+INT(Q/2)
1640 IF B=TT THEN 1670
```

```
1650 IF B>DIFF THEN 1670 ELSE 1820
1660 REM AVION DISPARA LASER
1670 IF BA=5 THEN 1760 ELSE 1680
1680 IF B=16 THEN 1682 ELSE 1760
1682 BA=BA+1
1683 FG$=STR$(BA)
1684 BB=ASC(SEG$(FG$,1,1))
1691 FF$="BARRERA: "
1692 FOR I=1 TO LEN(FF$)
1693 FK=ASC(SEG$(FF$,I,1))
1694 CALL HCHAR(AA+1,22+I,FK)
1695 NEXT I
1696 CALL HCHAR(AA+1,15,BB)
1700 IF BA<>5 THEN 1720
1710 CALL HCHAR(22,15,32,3)
1720 CALL VCHAR(A+1,B,144,21-A)
1730 CALL VCHAR(A+1,B,32,21-A)
1740 CALL SOUND(500,-5,2)
1750 GOTO 1850
1760 CALL VCHAR(A+1,B,144,23-A)
1770 IF B<>T+1 THEN 1790
1780 IF A<>Y+2 THEN 2440
1790 CALL VCHAR(A+1,B,32,22-A)
1800 CALL VCHAR(23,B,128,2)
1810 CALL SOUND(500,-5,2)
1820 IF B>30 THEN 1840 ELSE 1830
1830 IF B<3 THEN 1840 ELSE 1850
1840 CALL VCHAR(A,B,32)
1850 IF A<>Y+2 THEN 1870
1860 IF B=T+1 THEN 2050
1870 B=B+DIR
1880 IF B>32 THEN 1900 ELSE 1890
1890 IF K=81 THEN 1420 ELSE 1400
1900 RANDOMIZE
1910 B=INT(2*RND)+1
1920 REM DIRECCION AVION
1930 ON B GOTO 1940,1980
1940 B=30
1950 CALL CHAR(96,PR$)
1960 DIR=-2
1970 GOTO 2000
1980 DIR=2
1990 CALL CHAR(96,P$)
2000 RANDOMIZE
2010 A=2*INT(9*RND)+1
2020 IF A=1 THEN 2000
2030 GOTO 1890
2040 REM TANQUE TIRA
2050 CALL SOUND(1000,-5,2)
2060 REM
2070 CALL HCHAR(24,T+1,42)
2080 CALL HCHAR(23,T+1,152)
2090 CALL HCHAR(A,B,120)
2100 REM
2110 B=INT(2*RND)+1
2120 ON B GOTO 2130,2170
2130 B=30
2140 DIR=-2
2150 CALL CHAR(96,PR$)
2160 GOTO 2200
2170 DIR=2
2180 CALL CHAR(96,P$)
2190 REM * SCORE *
2200 IF A>13 THEN 2210 ELSE 2230
2210 SC=SC+5
2220 GOTO 2300
2230 IF A>7 THEN 2240 ELSE 2260
2240 SC=SC+10
2250 GOTO 2300
2260 IF A<>5 THEN 2290
2270 SC=SC+15
2280 GOTO 2300
2290 SC=SC+20
2300 M$=STR$(SC)
2310 RANDOMIZE
2320 A=2*INT(9*RND)+1
2330 IF A=1 THEN 2310
2340 REM SCORE
2341 QW$="PUNTAJE: "
2342 FOR I=1 TO LEN(QW$)
2343 QA=ASC(SEG$(QW$,I,1))
2344 CALL HCHAR(AA,22+I,QA)
2345 NEXT I
2350 FOR I=1 TO LEN(M$)
2360 CV=ASC(SEG$(M$,I,1))
2370 CALL HCHAR(AA,22+I+10,CV)
2380 NEXT I
2390 GOTO 1380
2400 FOR I=1 TO 1000
2410 NEXT I
2420 GOTO 1040
2430 REM AVION PEGA AL TANQUE
2440 CALL SOUND(500,-6,2)
2450 CALL CLEAR
2460 SX=SX+1
2470 REM DESPUES DE 2 EMPIEZA
2480 IF SX<2 THEN 1040
2490 PRINT TAB(7);"TERMINO EL JUEGO"
2500 PRINT TAB(7);"SU PUNTAJE ES";SC
2510 FOR I=1 TO 10
2520 PRINT
2530 NEXT I
2540 SX=0
2550 IF SC<100 THEN 2600
2560 PRINT TAB(3);"* GANO UN NUEVO JUEGO *"
2570 REM
2580 SC=0
2590 GOTO 2400
2600 SC=0
2610 REM
2615 FOR AR=1 TO 500
2616 NEXT AR
2620 GOTO 680
2630 REM
2640 IF T<3 THEN 2650 ELSE 2680
2650 CALL SOUND(250,110,2)
2660 T=1
2670 GOTO 2690
2680 T=T-2
2690 CALL HCHAR(24,T,112)
2700 CALL HCHAR(24,T+1,42)
2710 CALL HCHAR(24,T+2,113)
2720 CALL HCHAR(23,T+1,152)
2730 CALL HCHAR(24,T+3,128,2)
2740 CALL HCHAR(23,T+3,128)
2750 GOTO 1550
2760 REM
2770 IF T>27 THEN 2780 ELSE 2810
2780 CALL SOUND(250,110,2)
2790 T=29
2800 GOTO 2820
2810 T=T+2
2820 CALL HCHAR(24,T,112)
2830 CALL HCHAR(24,T+1,42)
2840 CALL HCHAR(24,T+2,113)
2850 CALL HCHAR(23,T+1,152)
2860 CALL HCHAR(24,T-2,128,2)
2870 CALL HCHAR(23,T-1,128)
2880 GOTO 1550
2890 GOTO 1040
2900 END
```

REVISION K64 SOFTWARE

Todos los meses, K64 publicará revisiones y tests realizados sobre el software que se consigue en nuestro medio para las computadoras que seguimos en esta revista.

Nuestro criterio apunta a la evaluación objetiva de la calidad, utilidad, presentación y costo del material recibido.

No habiendo ningún interés comercial de nuestra parte en esta evaluación, la intención es solamente guiar e informar al lector.

La revisión K64 comienza con un recuadro de referencia rápida tipo ficha, donde se indica resumidamente el puntaje relativo para tener en un golpe de vista, una primera aproximación de que se trata.

FOTO PANTALLA

Nombre:
Clasificación:
Computadora:
Configuración:
Distribuye:
Precio aprox.:
Factor K64:
Hechizo:
Documentación:

Factor K64:
Cómo se comporta el programa según promete, cómo explota la capacidad de la máquina, qué tan bien responde a los comandos del usuario, resultado de gráficos, efectos sonoros, música, o síntesis vocal que hayan sido integrados en el programa.

Hechizo:
Capacidad o virtud intangible del programa, que mantiene al jugador pegado en la silla mientras las horas pasan sin darse uno cuenta (aplicable sobre todo a programas de juegos y educativos).

Documentación:
Calidad del material impreso que acompaña al programa. Claridad y detalle en la explicación de los conceptos sobre el funcionamiento del programa y su utilidad.
Detalles de uso, carga y control entendibles aún por el inexperto.

Además se tendrá en cuenta:

Flexibilidad:
Puede adaptarse el producto a necesidades específicas del usuario?

Costo/Beneficio:
¿Es un programa caro por lo que hace?

Presentación:
Calidad de grabación, envase, etc.

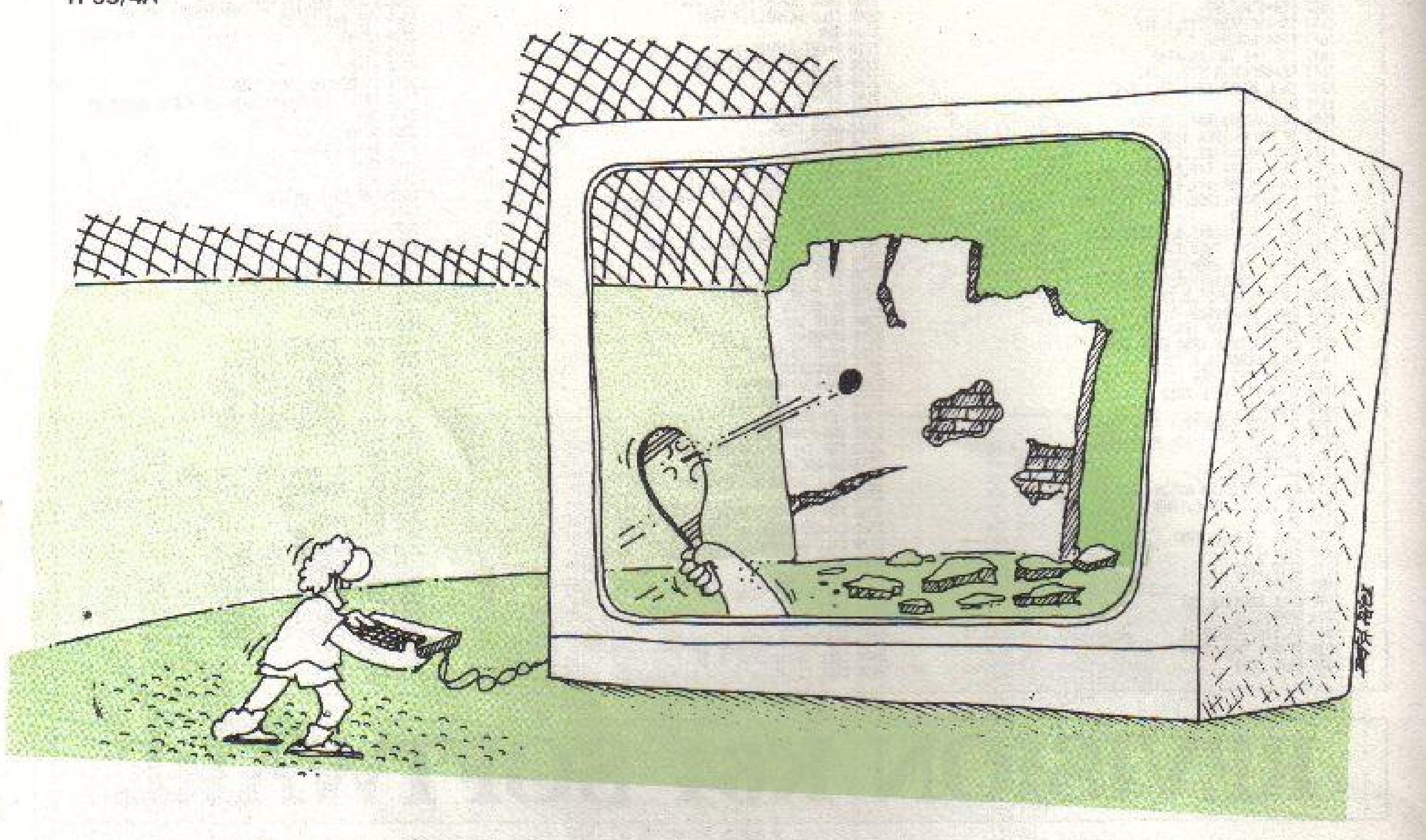
Utilidad:
¿El programa resuelve algún problema o necesidad realmente existente?

Trascendencia:
¿Mantiene interés o utilidad el programa a través del tiempo?

Efectos especiales:
¿Cómo se comportan los efectos visuales y sonoros en el contexto del programa?
¿Mejoran o ayudan a la comprensión del mismo o a su espectacularidad?

FRONTON

TI 99/4A



```
1 FOR Y=1 TO 10
2 CALL COLOR(Y,16,7)
3 NEXT Y
4 CALL CLEAR
5 CALL SCREEN(7)
6 CALL COLOR(9,2,7)
7 CALL CHAR(96,"00000000F0F0F0F0")
8 CALL CHAR(97,"F0F0F0F0")
9 CALL CHAR(98,"FFFFFFFFF0F0F0F0")
10 CALL CHAR(99,"F0F0F0F0F0F0F0F0")
11 CALL CHAR(100,"FFFFFFFFFFFFFFFF")
12 CALL CHAR(101,"FF00FF00FF00FF00")
13 CALL CHAR(102,"F000F000F000F000")
14 CALL VCHAR(4,3,100,2)
15 CALL VCHAR(4,6,100,2)
16 CALL VCHAR(4,9,100,2)
17 CALL HCHAR(5,4,96)
18 CALL HCHAR(4,4,97)
19 CALL HCHAR(3,3,98)
20 CALL HCHAR(3,4,97)
21 CALL HCHAR(5,5,96)
22 CALL HCHAR(3,6,98)
23 CALL HCHAR(3,7,99)
24 CALL HCHAR(4,7,97)
25 CALL HCHAR(5,8,96)
26 CALL HCHAR(3,9,99)
27 CALL HCHAR(5,11,96)
28 CALL HCHAR(2,9,97)
29 CALL HCHAR(24,1,101,30)
30 CALL VCHAR(1,30,102,24)
31 LABEL$="EMPRESA PARA INFORMATICA"
32 F=8
33 C=3
34 GOSUB 58
35 LABEL$="PROGRAMAS PARA EL"
36 F=12
37 C=3
38 GOSUB 58
39 LABEL$="MICROCOMPUTADOR TI-99/4A"
40 F=14
41 C=3
42 GOSUB 58
43 LABEL$="FABRICADO Y DISTRIBUIDO  EN"
44 F=18
45 C=3
46 GOSUB 58
47 LABEL$="LA REPUBLICA  ARGENTINA POR"
48 F=20
49 C=3
50 GOSUB 58
51 LABEL$="'SDT'INTELIGENCIA ARGENTINA"
52 F=22
53 C=3
54 GOSUB 58
55 FOR TT=1 TO 300
56 NEXT TT
57 GOTO 64
58 FOR T=1 TO LEN(LABEL$)
59 A$=SEG$(LABEL$,T,1)
60 B=ASC(A$)
61 CALL HCHAR(F,C-1+T,B)
62 NEXT T
63 RETURN
64 CALL CHARSET
100 CALL CLEAR
110 PRINT
120 PRINT "EN ESTE JUEGO USTEDES SOLO  TIENEN CONTROL
SOBRE LA PELOTA CON LAS TE
CLAS S Y D."
130 PRINT "DEBERAN GUIARLA PARA QUE PE-GUE EN LA
PALETA QUE POR SU-PUESTO TRATA
DE EVITARLO."
140 PRINT "ADEMAS, CUANDO LA PELOTA SU-BA,DEBEN TRATAR
DE DESTRUIR LO MAS QUE PU
EDAN AL FRONTONY ASI LOGRARAN MAS PUNTOS."
150 PRINT :: PRINT :: PRINT
151 REM DISPLAY AT(12,2):"OTRO JUEGO ? (S=ENTER/N=NO)."
160 GOTO 660
170 !########################
180 !   P A R E D O N
190 !########################
200 CALL CHAR(100,"FFFFFFFFFFFFFFFF")
210 CALL CHAR(101,"CC33CC33CC33CC33")
220 !####################
230 !BORRAR PANTALLA  ARMAR      CIRCULO
231 CALL SCREEN(16)
232 CALL COLOR(9,5,1)
240 CALL CLEAR :: RANDOMIZE
250 FOR A=1 TO 100
260 LET B=PI*A/50
270 XX=9*COS(B)+12
280 YY=14*SIN(B)+17
290 CALL HCHAR(XX,YY,100)
300 NEXT A
310 !####################
320 ! COLOCAR OBSTACULOS
330 !####################
340 FOR O=5 TO 8 :: FOR O1=11 TO 22 :: CALL HCHAR
(O,O1,101):: NEXT O1 :: NEXT O
350 !####################
360 LET C,D=10 :: SCORE=0 :: A=19 :: B=13 :: E=1 ::
K=5 :: A1=19 :: B1=21 :: C1,
D1=10
370 REM
380 LET B=B-RND*4+RND*4
390 CALL KEY(1,KK,KS)
400 IF KK=2 THEN LET D=D-1
410 IF D<7 THEN LET D=7
420 IF KK=3 THEN LET D=D+1
430 IF D>24 THEN LET D=24
440 IF B>22 THEN LET B=22
450 IF B<9 THEN LET B=9
460 LET C=C+E
465 DISPLAY AT(1,1)SIZE(6):"GOLPES"
468 DISPLAY AT(1,22)SIZE(6):"PUNTOS"
470 DISPLAY AT(2,2)SIZE(3):K
480 CALL HCHAR(A1,B1,32,3)
490 CALL HCHAR(C1,D1,32)
500 CALL HCHAR(A,B,30,3)
510 CALL HCHAR(C,D,30)
520 IF C=18 THEN IF ABS(D-(B+1))>1 THEN
 LET K=K-1 :: GOSUB 650
530 IF C+1=19 THEN LET E=-1
540 A1=A :: B1=B :: C1=C :: D1=D
550 IF K=0 THEN 590
560 IF C=5 THEN LET E=1
570 LET SCORE=SCORE+1
580 GOTO 370
590 FOR O=5 TO 8 :: FOR O1=11 TO 22 ::
CALL GCHAR(O,O1,CAR)
600 IF CAR=32 THEN SC=SC+1
610 NEXT O1 :: DISPLAY AT(2,24)SIZE(3):SC :: NEXT O
630 FOR TT=1 TO 500 :: NEXT TT
640 RUN 160
650 FOR SON=1110 TO 110 STEP -20 :: CALL SOUND
(-100,SON,3,-3,2):: NEXT SON :: RE
TURN
660 DISPLAY AT(23,1):"OPRIMA ENTER PARA JUGAR,
"N" PARA FINALIZAR"
670 CALL KEY(0,K,S)
671 IF S=0 THEN 670
680 IF K=13 THEN 200
682 IF K=78 THEN END
683 IF K<>13 AND K<>73 THEN 670
```

CONOCIENDO LAS COMPUTADORAS

Vamos a ver un poco cómo se organizan las unidades de disco normalmente usadas en las micro.

Para comprender el funcionamiento de una unidad de disco debemos pensar en dos bloques distintos:

1- Rutinas de grabación y lectura de sectores.

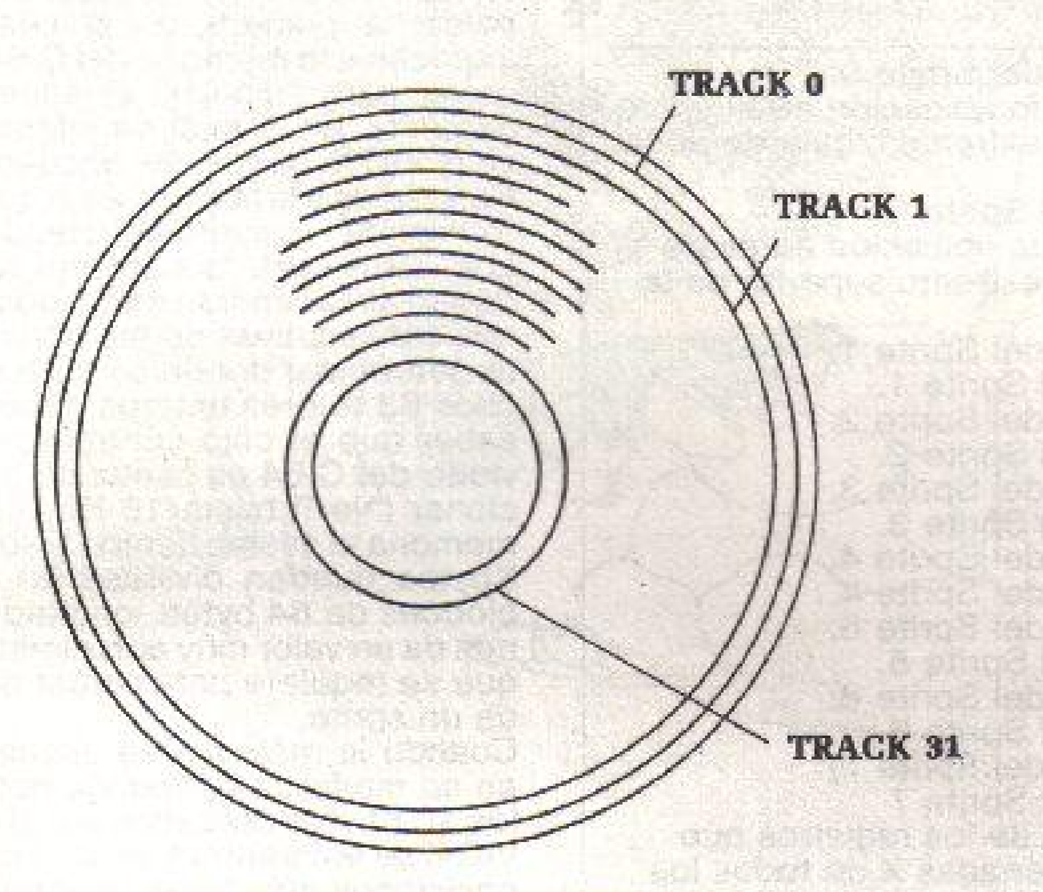
2- Sistema operativo de disco (D.O.S.).

El disco desde el punto de vista del hardware puede ser considerado como un aparato capaz de grabar información en unos casilleros identificados con dos índices: el "track" o pista y el sector, o sea una matriz de casilleros de dos dimensiones. Cada casillero tiene una capacidad normalmente de 128 ó 256 bytes. La mínima cantidad de información que el disco puede leer o grabar es justamente el tamaño del "casillero". Normalmente existen dos subrutinas en lenguaje de máquina que permite, una grabar un "casillero", para ésto tenemos que indicarle en qué sector y en qué pista queremos que grabe y la dirección de comienzo del "buffer" donde tenemos la información a grabar; la otra rutina lee un "casillero" debiéndole suministrársele el track, sector y dirección inicial donde queremos que ponga la información que lee. En los manuales del sistema operativo a veces se informa sobre el uso de estas rutinas directamente. Estas rutinas pueden ser muy complicadas y con cálculos de tiempos muy delicados en caso que la implementación del disco se haya hecho sin "disk controller" o sea todo por software.

Un disk controller es normalmente un chip (o algunos pocos) que se encargan directamente de grabar o leer en el track y sector que nosotros le indicamos, para ejecutar ésto debe hacer un acceso directo a memoria (DMA) que es una operación por la cual toma el control de los buses y anula al procesador el cual se queda esperando hasta que se completa la lectura o grabación para luego retomar el control, esto se hace para que el controller lea o escriba en el buffer que tenemos en la memoria. Estos controllers tienen la ventaja de su velocidad y simplicidad para el que arma una micro pero agregan un costo que a veces es preferible evitar poniendo más software.

Claro que si un disco se usaría como lo hemos descripto hasta ahora sería muy engorroso porque deberíamos llevar la cuenta de cuáles casilleros tenemos ocupados o si un programa ocupa más de un casillero; tendríamos que saber cuál es el primer casillero, luego el siguiente y así hasta el último, porque un programa grande no está necesariamente grabado en casilleros contiguos (aunque sí está dispuesto en cierto orden el tiempo de carga disminuye). De todas estas cosas y más se encarga el sistema operativo de disco siendo una de las más importantes el mantenimiento del directorio del disco que nos indica qué tenemos guardado en el disco y en qué posiciones se encuentra para poder leerlo en orden (generalmente el directorio es un archivo más aunque con una posición fija en el disco para acelerar la lectura).

Una aclaración importante es que son físicamente el track o pista y el sector en un disco. Para ello mostramos en la figura su disposición:



DIVISION DEL DISCO EN TRACKS



DIVISION DEL TRACK EN SECTORES

Aquí vemos que el disco está dividido en circunsferencias concéntricas las cuales a su vez están divididas en sectores, las circunsferencias se llaman tracks estando numeradas siendo 0 la más grande y 31 la más chica (en el ejemplo) y cada track está dividido en 16 sectores del 0 al 15.

En el caso de la Commodore 64 (R) todo esto que contamos está un poco oculto ya que la unidad de disco tiene su propio microprocesador y el intercambio de información se hace a través de un bus serie especial.

MARCELO OSCAR MARTINEZ

MOVIENDO SPRITES

En nuestra última edición proporcionamos a los lectores un programa para el diseño de objetos móviles de alta resolución (24 x 21 puntos) más comúnmente llamados "sprites". Si el usuario tenía alguna experiencia en la programación de sprites en su sistema C-64, estamos seguros que AM-SPRITER habrá sido de gran utilidad para él. Pero si el lector era un programador novicio, dicho programa habrá significado un gran esfuerzo de tipeado y magros resultados al correrlo.
Por esta razón, hoy nos dedicaremos a estudiar varios registros del chip interfase de video VIC-II, que se encargan de manejar estos bloques móviles. Es una buena medida que el lector tenga a mano la cartilla de la página 176 (Easy Spritemaking Chart) de la versión inglesa de la Guía de Referencia del Programador o bien el apéndice O de la Guía del Usuario (Página 159 en la versión inglesa), ya que ambas son de por sí ilustrativas sobre este tema.

REGISTROS DE SPRITES DEL CHIP VIC-II:

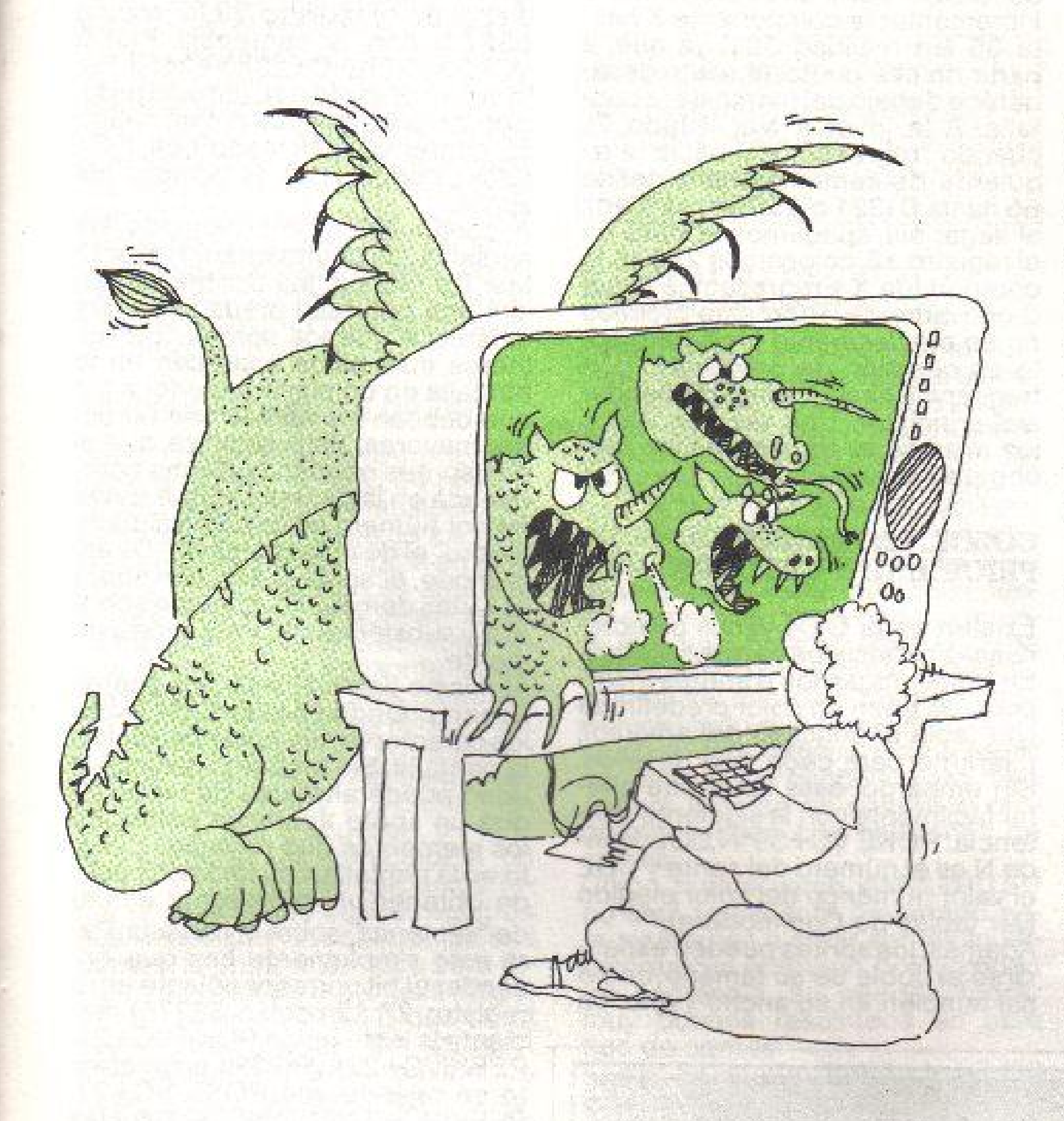
El área RAM reservada para el chip de video VIC-II comienza en la dirección decimal 53248 y abarca un total de 47 registros de 1 byte (8 bits) de longitud cada uno. A continuación, veamos un listado de registros relacionados con el manejo de sprites, cuyas funciones veremos en detalle en el presente artículo:

POSICION DE MEMORIA	CONTENIDO
53248 (registro 0)	Posición horizontal del Sprite 0. Este registro indica la ubicación en el eje X, comenzando por el extremo izquierdo de la pantalla.
53249 (registro 1)	Posición vertical del Sprite 0. Este registro indica la ubicación en el eje Y, comenzando por el extremo superior de la pantalla.
53250 (registro 2)	Posición horizontal del Sprite 1.
53251 (registro 3)	Posición vertical del Sprite 1.
53252 (registro 4)	Posición horizontal del Sprite 2.
53253 (registro 5)	Posición vertical del Sprite 2.
53254 (registro 6)	Posición horizontal del Sprite 3.
53255 (registro 7)	Posición vertical del Sprite 3.
53256 (registro 8)	Posición horizontal del Sprite 4.
53257 (registro 9)	Posición horizontal del Sprite 4.
53258 (registro 10)	Posición horizontal del Sprite 5.
53259 (registro 11)	Posición vertical del Sprite 5.
53260 (registro 12)	Posición horizontal del Sprite 6.
53261 (registro 13)	Posición vertical del Sprite 6.
53262 (registro 14)	Posición horizontal del Sprite 7)
53263 (registro 15)	Posición vertical del Sprite 7
53264 (registro 16)	Bit más significativo de los registros que contienen las coordenadas X de todos los sprites.
53269 (registro 21)	Registro de encendido de los sprites.
53271 (registro 23)	Registro de expansión de sprites en el eje Y.
53275 (registro 27)	Registro de prioridad de sprites sobre el fondo de la pantalla.
53276 (registro 28)	Registro designador de sprites multicolor.
53277 (registro 29)	Registro de expansión de sprites en el eje X.
53278 (registro 30)	Registro indicador de colisión entre sprites.
53279 (registro 31)	Registro indicador de colisión entre sprites y el fondo de la pantalla.
53285 (registro 37)	Código de color número 0 del sprite multicolor. Predefinido en verde.
53286 (registro 38)	Código de color número 1 del sprite multicolor. Predefinido en azul.
53287 (registro 39)	Color del sprite 0. Predefinido en amarillo.
53288 (registro 40)	Color del sprite 1. Predefinido en naranja.
53289 (registro 41)	Color del sprite 2. Predefinido en marrón.
53290 (registro 42)	Color del sprite 3. Predefinido en rojo claro.
53291 (registro 43)	Color del sprite 4. Predefinido en gris 1.
53292 (registro 44)	Color del sprite 5. Predefinido en gris 2.
53293 (registro 45)	Color del sprite 6. Predefinido en verde claro.
53294 (registro 46)	Color del sprite 7. Predefinido en azul claro.

DEFINIENDO EL SPRITE:

El proceso de definición de un sprite consiste de tres fases principales: la primera, en la cual se ingresan a la memoria del C-64 los datos para "dibujar" el sprite; la segunda, en la cual se informa al chip VIC-II donde se encuentran esos datos y la tercera, en la que se "enciende" el sprite así creado.
Los datos para "dibujar" un sprite deben almacenarse en 63 posiciones consecutivas de memoria. Para determinar donde se colocarán esos 63 valores enteros debemos saber que el chip controlador de video del C-64 es capaz de direccionar ("ver") hasta 16 Kbytes de memoria al mismo tiempo. Esos 16 Kbytes pueden dividirse en 250 bloques de 64 bytes, longitud que nos da un valor muy aproximado al que se requiere para cargar datos de un sprite.
Cuando la máquina se encuentra en su modo de operación normal, los 16 Kbytes utilizados por el chip VIC-II se encuentran en las 16.000 posiciones más bajas de memoria de la misma. Dentro de este sector, el primer bloque de 64 bytes de longitud ocupa las posiciones 0 a 63, el segundo las 64 a 127 y así sucesivamente. Si bien existen 250 bloques en esta área, no todos ellos están disponibles al usuario ya que una gran cantidad de bytes de este sector están reservados por el sistema operativo Kernal mientras que 1.000 bytes adicionales están reservados para los caracteres que se colocan en la pantalla.
No obstante ello, existen en este sector algunos bloques que pueden ser utilizados sin correr el riesgo de perturbar los contenidos de la pantalla y de los diversos registros Kernal. Si el programa no utiliza más de cuatro sprites de distinto diseño, sus datos pueden colo-

EN LA PANTALLA



carse en los bloques números 11 (que comienza en 704), 13 (en 832), 14 (en 896) y 15 (en 960). Si se deben emplear más de 4, el Manual de Referencia del Programador recomienda utilizar los bloques de 192 a 199 que comienzan en la posición 12288. Colocar sprites en este sector, no obstante, puede perturbar la RAM BASIC del sistema, por lo que se debe tener mucho cuidado al programar juegos muy extensos con muchos sprites, ya que la carga de datos en este sector pueden alterar el programa mismo.

Con la ayuda de AM-SPRITER (ver K64 número 3, página 26), entonces, generamos los datos numéricos de un sprite y los almacenamos ya sea en un archivo en disco o casette, o en una serie de sentencias DATA de nuestro programa de sprites. A continuación, un ciclo FOR debe leer esos valores y POKEarlos en alguno de los bloques citados en el párrafo anterior. Con esto termina la primera etapa de la definición del sprite.

La segunda etapa del proceso de definición del sprite consiste en indicarle al chip VIC-II donde debe buscar los datos que componen el sprite. A partir de la dirección 2040 existen ocho registros llamados "punteros de los sprites" (sprites pointers) cuyos contenidos indican los números de bloque en los cuales se encuentran los datos que "dibujan" cada sprite. Así, el registro 2040 posee el puntero al sprite 0, el 2041 el puntero al sprite 1, etc. Una vez cargados los datos del sprite y asignado su correspondiente puntero, sólo falta encenderlo, es decir, ejecutar la tercera etapa de la definición. El registro VIC-II número 21 (dirección 53269) se asemeja a un "tablero" de 8 interruptores cada uno de los cuales controla el "encendido" (aparición en la pantalla) de un sprite en particular. De este modo, colocando en 1 el bit número 0 se enciende el sprite 0 mientras que un 1 en el bit número 1 enciende el sprite 1; y así sucesivamente. Para encender un sprite se debe ejecutar la sentencia POKE 53269,PEEK(53269)OR (2↑N), donde N es el número de sprite. De forma análoga se debe ejecutar la sentencia POKE 53269, PEEK(53269)AND(NOT(2↑N) para apagar dicho sprite.

MOVIMIENTO DEL SPRITE EN LA PANTALLA:

El principio de posicionamiento y desplazamiento de cada sprite en la pantalla se basa en tres registros del chip VIC-II. Uno contiene la coordenada X (posición en el eje horizontal), otro la coordenada Y (posición en el eje vertical) y el tercero (registro 16, dirección 53264) tiene un bit adicional, llamado "bit más significativo" (BMS), para el registro de la coordenada X. Para cada sprite, la coordenada X se encuentra en la dirección VC+2*N mientras que la coordenada Y está en la dirección VC+2*N+1, donde VC equivale a 53248 (el principio del chip VIC-II) y N indica el número del sprite. Por ejemplo, el registro 0 contiene la componente X del sprite y el registro 1 su coordenada Y. Para ubicar un sprite en la pantalla, se deben tener en cuenta algunos factores. En primer lugar, se debe tener como referencia que las coordenadas X=0, Y=0 señalen a un punto en el EXTREMO SUPERIOR IZQUIERDO de la pantalla. Además, existen ciertos valores de los registros que contienen las componentes X e Y con los cuales el sprite no puede verse ya que el mismo queda oculto debajo del "marco". Esto sucede cuando la coordenada Y es menor que 50 o mayor que 250; y cuando la componente X es menor que 24.

Por otro lado, un registro de 8 bits resulta insuficiente para posicionar el sprite en la pantalla ya que el mismo sólo puede contener valores acotados entre 0 (0000 0000 en binario) y 255 (1111 1111); mientras que la pantalla tiene 320 puntos de ancho. Para solucionar este problema se emplea el registro 16, que contiene el BMS de la coordenada X de cada sprite. Si, alcanzado el valor 255 (aproximadamente 3/4 de la pantalla), se desea desplazar el sprite más a la derecha, se debe encender el correspondiente BMS mediante la sentencia POKE VC+16,PEEK(VC+16) OR(2↑N) donde VC es igual a 53248 y N el número del sprite; mientras que para apagar dicho BMS debemos ejecutar la sentencia POKE VC+16,PEEK(VC+16)AND(NOT(2↑N).

Agregando el BMS al registro de la coordenada X podemos obtener un conjunto de valores acotados entre 512 (1 0000 0000 binario) y 768 (1 1111 1111 binario) que nos permiten desplazarnos por el 1/4 restante a la derecha de la pantalla. Se debe tener precaución al encender el BMS de una coordenada X ya que, a partir de ese momento, debemos volver a modificar el contenido del registro de dicha componente.

Supongamos, por ejemplo, que deseamos mover el sprite N por una línea horizontal cualquiera. Para ello, ejecutamos el breve programa incluido en el listado 1, que nos lleva desde el extremo izquierdo al extremo derecho. Ahora bien, cuando deseamos desplazarnos en la misma línea horizontal, pero de derecha a izquierda, corremos el programa del listado 2.

Nótese que en el listado 1, al llegar a la posición X=255 debemos encender el BMS del sprite N y recargar VC+2*N con el valor 0 (en realidad 256 ya que el BMS está encendido). A partir de allí, volvemos a incrementar la componente X hasta 65 (en realidad 321) ya que, a partir de ese punto, el sprite desaparece debajo del marco de la pantalla. A la inversa (ver listado 2), cuando "regresamos'6 hacia la izquierda decrementamos X desde 65 hasta 0 (321 a 256 en realidad); al llegar allí, apagamos el BMS en el registro 16, colocamos 255 en la coordenada X y regresamos hasta 0 nuevamente. Todo este proceso no es precisamente fácil; pero sólo es cuestión de practicarlo con frecuencia hasta lograr aprehender sus principios. Una vez conocidos los mismos, el trabajo resulta mucho más simple.

COLOR, EXPANSION, PRIORIDAD Y COLISION:

Existen en el C-64 varias características especiales para los sprites. En primer lugar, los 8 sprites incorporados traen un color predefinido (ver tabla al principio del artículo); diferente para cada uno de ellos. Sin embargo, este se puede alterar fácilmente con la siguiente sentencia: POKE VC+39*N,COL; donde N es el número del sprite y COL el valor numérico del color elegido (ver tablas de Commodore).

Además, los sprites pueden expandirse al doble de su tamaño, como así también en su ancho y alto. La expansión en el eje X se efectúa mediante el encendido del bit correspondiente en el registro 29 (dirección 53277) con la ya tradicional fórmula: POKE VC+29,PEEK(VC+29)OR(2↑N). A su vez, el sprite se expande en el eje Y por medio del encendido del bit correspondiente en el registro 29 (dirección 53271) con la sentencia POKE VC+23,PEEK(VC+23)OR(2↑N). Para volver el sprite a su tamaño reducido, ambos registros deben recibir su contenido ANDeado con NOT (2↑N), donde N es el número del sprite.

A continuación, encontramos las así llamadas prioridades de los sprites. De fábrica, los sprites vienen con una prioridad predefinida que establece que los sprites con números más bajos aparecen en la pantalla en un plano superior a los que ocupan los sprites con números mayores. Esto significa que si dos sprites ocupan la misma coordenada en la pantalla, el que tenga menor número se ve como situado "sobre" el de número mayor. De este modo, el sprite 0 aparece sobre el 1 y los demás, el 1 aparece sobre el 2 y subsiguientes y así sucesivamente.

Al mismo tiempo, todos los sprites tienen prioridad predefinida sobre los caracteres que aparecen sobre la pantalla. Sin embargo, esta prioridad puede alterarse de modo tal que un sprite aparezca detrás de los elementos que hayamos puesto en la pantalla, con lo cual se puede obtener un interesante efecto de "ventanas" sobre la pantalla. Para esto simplemente hay que encender el bit correspondiente en el registro 27 (dirección 53275) mediante la instrucción POKE VC+27, PEEK(VC+27)OR(2↑N), cuyo efecto se revierte con POKE VC+27, PEEK(VC+27)AND(NOT(2↑N). Hay que ser muy precavidos al cambiar esta prioridad ya que si, por ejemplo, hacemos que el sprite 0 tenga prioridad menor a la del fondo y que el sprite 1 tenga mayor prioridad que el mismo, tendremos un interesante caso de incoherencia geométrica ya que el sprite 0 conservará su prioridad sobre el sprite 1 (es decir que pasará sobre él) pero, al mismo tiempo, pasará debajo del fondo de la pantalla, que tendrá menor prioridad que el sprite 1!!

Finalmente, existen dos registros que se encargan de informarnos si han ocurrido colisiones entre dos sprites o entre un sprite y el fondo de la pantalla. Si sucede una colisión se encenderán diversos bits y, leyendo dichos registros, determinaremos cuál o cuáles sprites han estado involucrados. Se debe te-

ner especial precaución al leer con un PEEK estos registros ya que una vez leídos, su contenido es restaurado a 0. Por tal motivo es altamente recomendable guardar el contenido de esos registros en alguna variable.

En el caso de una colisión entre dos sprites, el registro a leer es el número 30 (dirección 53278) mientras que cuando un sprite choca contra el fondo se debe inspeccionar el registro 31 (dirección 53279). El modo por el cual se determina qué sprites estuvieron involucrados en la colisión varía notablemente según la cantidad de sprites empleados y el tipo de programa; pero generalmente se puede resolver mediante diversas operaciones AND.

SPRITES MULTICOLORES:

Concluyendo esta revisión del manejo de sprites en el C-64 nos referiremos a los sprites multicolores. Estos son una variante de los sprites comunes ya descriptos en este artículo y sólo difieren de aquéllos en que sus puntos pueden tener hasta cuatro colores distintos y en que su resolución horizontal se reduce a la mitad. En otras palabras, los sprites multicolores miden 12 x 21 pixels contra 24 x 21 de sus hermanos normales. Por lo demás, se definen, apuntan y posicionan en forma idéntica a aquéllos.

Al incorporar un miltucolor a un programa de sprites debemos tener en cuenta los cuatro registros que determinan los colores de sus puntos. El primero de ellos es el registro de color de la pantalla (registro 33, posición 53281) que da el color "transparente" del sprite. A continuación está el registro de color propio del sprite que se designa como multicolor (registros 39 a 46) y dos registros adicionales comunes a todos los sprites multicolores, los registros 37 y 38 (direcciones 53285 y 53286), también llamados registros de sprite multicolor número 0 y número 1.

Si bien el sprite multicolor pierde el 50% de definición horizontal, éste sigue estando definido por un área de 63 bytes. En cada uno de estos bytes, los bits se agrupan de a 2 para dar la configuración de un solo punto. Según dicha configuración, el color de cada punto del sprite se determina de acuerdo a la siguiente tabla:

BITS:	COLOR ADOPTADO
01:	Multicolor 0. Registro 37
10:	Color Sprite. Registro 39+N
11:	Multicolor 1. Registro 38

N=NUMERO DEL SPRITE DESIGNADO MULTICOLOR:

Para designar un sprite como multicolor, se debe encender el bit correspondiente en el registro 28 (posición 53276) mediante la operación POKE VC+28,PEEK(VC+28) OR(2↑N), cuyos efectos se anulan con POKE VC+28,PEEK(VC+28) AND(NOT(2↑N). Una vez establecidos todos los parámetros aquí descriptos, el sprite multicolor puede manejarse como uno cualquiera del tipo normal. Vale la pena destacar que en el caso de las colisiones, los puntos cuyo color está determinado por el par de bits 01 se considera transparente a pesar de verse en la pantalla. Este dato deberá tenerse en cuenta al programar juegos con colisiones de sprites.

Como puede verse, AM-SPRITER no está totalmente preparado para ayudarnos a crear sprites multicolores. Sin embargo, con un poco de paciencia, se pueden lograr muy buenos resultados en este tipo de sprites.

Si alguno de nuestros lectores llegara a tener dificultad para comprender los conceptos comentados en esta apretada síntesis, le rogamos escriba a **K64 Computación para todos** para hacernos conocer sus dudas, las que trataremos de responder en futuras ediciones del Bus del Commodore 64.

CARLOS AY y DANIEL MANDUCA

```
1000 REM ******************************
1001 REM LISTADO EJEMPLO 1
1002 REM ******************************
1003 :
1004 :
1010 REM N=NUMERO DEL SPRITE
1020 REM VC=DIRECCION DEL VIDEO CHIP
1021 :
1022 :
1030 POKE VC+N*2+1,100:REM COORD. Y
1040 POKE VC+2*N,0:REM COORD. X INICIAL
1050 FOR X=0 TO 255
1060 POKE VC+2*N,X
1070 NEXT X
1080 REM ENCIENDE BIT MAS SIGNIFICATIVO
1090 POKE VC+16,PEEK(VC+16)OR(2↑N)
1100 REM VUELVE A CARGAR COORD. X
1110 POKE VC+2*N,0
1120 FOR X=0 TO 65
1130 POKE VC+2*N,X
1140 NEXT X
1150 END

READY.
```

```
1000 REM ******************************
1001 REM LISTADO EJEMPLO 2
1002 REM ******************************
1003 :
1004 :
1010 REM N=NUMERO DEL SPRITE
1020 REM VC=DIRECCION DEL VIDEO CHIP
1021 :
1022 :
1030 REM ENCIENDE BIT MAS SIGNIFICATIVO
1040 POKE VC+16,PEEK(VC+16)OR(2↑N)
1050 REM COORDENADA Y
1060 POKE VC+2*N+1,100
1070 REM COORDENADA X INICIAL
1080 POKE VC+2*N,65
1090 REM DESPLAZA HACIA LA IZQUIERDA
1100 FOR X=65 TO 0 STEP -1
1110 POKE VC+2*N,X
1120 NEXT X
1130 REM APAGA BIT MAS SIGNIFICATIVO
1140 POKEVC+16,PEEK(VC+16)AND(NOT(2↑N)
)
1150 REM RECARGA COORDENADA X
1160 POKE VC+2*N,255
1170 TERMINA DESPLAZAMIENTO
1180 FOR X=255 TO 0 STEP -1
1190 POKE VC+2*N,X
1200 NEXT
1210 END

READY.
```

```
1 REM ***********************************
2 REM *                                 *
3 REM *   PROGRAMA DEMO SPRITE K-64     *
4 REM *                                 *
5 REM * COPYRIGHT 1985, CARLOS A. AY.   *
6 REM *                                 *
7 REM ***********************************
8 PRINT"{CLR}{CYAN}";:GOTO1000
10 REM LEE DATOS DEL SPRITE
11 :
20 DC=BL*64
30 FOR I=0 TO 62
40 READ A
50 POKE DC+I,A
60 NEXT I
70 RETURN
71 :
80 REM CENTRA TITULOS
81 :
90 CO=INT((40-LEN(MS$))/2)
100 PRINT"{CLR}";
110 PRINT TAB(CO);MS$
120 RETURN
121 :
1000 REM PRINCIPIO DEL DEMO
1001 :
1010 MS$="CARGANDO DATOS DE SPRITES"
1020 GOSUB 80
1030 VC=53248:REM PPIO. CHIP DE VIDEO
1040 POKE VC+32,6:REM COLOR DEL MARCO
1050 POKE VC+33,6:REM COLOR DEL FONDO
1051 :
1060 REM CARGA DATOS DE LOS SPRITES
1061 :
1070 BL=11:GOSUB 10
1080 BL=13:GOSUB 10
1090 BL=14:GOSUB 10
1100 BL=15:GOSUB 10
1110 BL=195:GOSUB 10
1120 BL=196:GOSUB 10
1130 BL=197:GOSUB 10
1140 BL=198:GOSUB 10
1150 BL=199:GOSUB 10
1151 :
1160 REM ASIGNA PUNTEROS
1161 :
1170 POKE 2040,11
1180 POKE 2041,13
1190 POKE 2042,14
1200 POKE 2043,15
1210 POKE 2044,195
1220 POKE 2045,196
1230 POKE 2046,197
1240 POKE 2047,198
1241 :
1250 REM ENCIENDE SPRITES
1251 :
1260 FOR SPRITE=0 TO 7
1270 POKE VC+21,PEEK(VC+21)OR(2↑SPRITE
)
1280 NEXT SPRITE
1281 :
1290 REM POSICIONA X INICIALES
1291 :
1300 FOR I=VC TO VC+14 STEP 2
1310 POKE I,24
1320 NEXT I
1321 :
1330 REM POSICIONA Y INICIALES
1331 :
1340 IN=58
1350 FOR I=VC+1 TO VC+15 STEP 2
1360 POKE I,IN
1370 IN=IN+22
1380 NEXT I
1381 :
1390 REM PRUEBA DE COLORES
1391 :
1400 MS$="PRUEBA DE COLORES"
1410 GOSUB 80
1420 FOR I=0 TO 4
1430 FOR SP=0 TO 7
1440 FOR CL=0 TO 15
1450 PIKE VC+39+SP,CL+SP
1460 IF I=4 THEN POKE VC+39+SP,SP+7
1470 NEXT CL
1480 NEXT SP
1490 NEXT I
1491 :
1500 RLM PRUEBA PRIORIDADES E/SPRITES
1501 :
1510 MS$="PRIORIDADES ENTRE SPRITES"
1520 GOSUB 80
1530 FOR SP=7 TO 1 STEP -1
1540 P0=VC+SP*2+1
1550 FOR I=PEEK(P0) TO 58 STEP -1
1560 POKE P0,I
1570 NEXT I
1580 FOR R=0 TO 100
1590 NEXT R
1600 NEXT SP
1601 :
1610 FOR R=0 TO 1000
1620 NEXT R
1621 :
1630 FOR SP=0 TO 6
1640 P0=VC+SP*2
1650 FOR I=24 TO ((7-SP)*24)+50
1660 POKE P0,I
1670 NEXT I
1680 FOR R=0 TO 200
1690 NEXT R
1700 NEXT SP
1701 :
1710 REM PRIORIDADES S/FONDO
1711 :
1720 MS$="PRIORIDADES SOBRE EL FONDO"
1730 GOSUB 80
1740 FOR I=0 TO 5
1750 PRINT
1760 NEXT I
1770 FOR I=1 TO 5
1780 FOR J=0 TO 39
1790 PRINT"{RVON} ";
1800 NEXT J
1810 FOR J=0 TO 39
1820 PRINT"{RVOF} ";
1830 NEXT J
1840 NEXT I
1850 FOR SP=0 TO 7
1860 POKE VC+27,(2↑SP)
1870 FOR Y=58 TO 229
1880 POKE VC+SP*2+1,Y
1890 NEXT Y
1900 FOR Y=229 TO 58 STEP -1
1910 POKE VC+SP*2+1,Y
1920 NEXT Y
1930 NEXT SP
1931 :
1940 REM PRUEBA SPRITE MULTICOLOR
1941 :
1950 MS$="SPRITE MULTICOLOR"
1960 GOSUB 80
1970 POKE VC+21,0:REM APAGA TODOS
1980 POKE 2040,199:REM NUEVO SPRITE 0
1990 POKE VC+37,1:REM MC COLOR 0
2000 POKE VC+38,15:REM MC COLOR 1
2010 POKE VC+39,14:REM COLOR SPRITE
2020 POKE VC+28,1:REM SPRITE 0 ES MC
2030 POKE VC,100
2040 POKE VC+1,100
2050 POKE VC+21,1:REM ENCIENDE SPRITE 0
2060 FOR X=100 TO 255
2070 POKE VC,X
2080 NEXT X
2090 FOR Y=100 TO 229
2100 POKE VC+1,Y
2110 NEXT Y
2120 POKE VC+16,1:POKE V,0
2130 FOR X=0 TO 65
2140 POKE VC,X
2150 NEXT X
2160 FOR Y=229 TO 58 STEP -1
2170 POKE VC+1,Y
2180 NEXT Y
2190 FOR X=65 TO 0 STEP -1
2200 POKE VC,X
2210 NEXT X
2220 POKE VC+16,0:POKE V,255
2230 FOR X=255 TO 100 STEP -1
2240 POKE VC,X
2250 POKE VC+1,PEEK(VC+1)+1
2260 NEXT X
2270 POKE VC+21,255
2280 FOR X=24 TO 255
2290 POKE VC+14,X
2300 NEXT X
2400 FOR Y=PEEK(VC+1) TO 58 STEP -1
2450 POKE VC+1,PEEK(VC+1)-1
2460 NEXT Y
2470 FOR X=PEEK(VC) TO 24 STEP -1
2480 POKE VC,PEEK(VC)-1
2490 NEXT X
2500 PRINT"{C/DN}{C/DN}{C/DN}"
2510 INPUT"OTRA VUELTA";RE$
2520 IF LEFT$(RE$,1)<>"S"THEN SYS64738
2530 POKE VC+28,0
2540 RUN
2541 :
2550 REM DATOS DE LOS SPRITES
2551 :
2560 DATA 255,255.255,225,255,255,204
2570 DATA 255,255,204,255,255,204,255
2580 DATA 255,204,255,255,204,255,255
2590 DATA 204,255,255,204,255,255,225
2600 DATA 255,255,255,255,255,255,255
2610 DATA 255,255,255,255,255,255,255
2620 DATA 255,255,255,255,255,255,255
2630 DATA 255,255,255,255,255,255,255
2640 DATA 255,255,255,255,255,255,255:
2650 DATA 255,255,255,227,255,255,195
2660 DATA 255,255,243,255,255,243,255
2670 DATA 255,243,255,255,243,255,255
2680 DATA 243,255,255,225,255,255,255
2690 DATA 255,255,255,255,255,255,255
2700 DATA 255,255,255,255,255,255,255
2710 DATA 255,255,255,255,255,255,255
2720 DATA 255,255,255,255,255,255,255
2730 DATA 255,255,255,255,255,255,255:
2740 DATA 255,255,255,225,255,255,204
2750 DATA 255,255,252,255,255,252,255
2760 DATA 255,192,255,255,207,255,255
2770 DATA 207,255,255,207,255,255,192
2780 DATA 255,255,255,255,255,255,255
2790 DATA 255,255,255,255,255,255,255
2800 DATA 255,255,255,255,255,255,255
2810 DATA 255,255,255,255,255,255,255
2820 DATA 255,255,255,255,255,255,255:
2830 DATA 255,255,255,225,255,255,204
2840 DATA 255,255,252,255,255,252,255
2850 DATA 255,193,255,255,252,255,255
2860 DATA 252,255,255,204,255,255,225
2870 DATA 255,255,255,255,255,255,255
2880 DATA 255,255,255,255,255,255,255
2890 DATA 255,255,255,255,255,255,255
2900 DATA 255,255,255,255,255,255,255
2910 DATA 255,255,255,255,255,255,255:
2920 DATA 255,255,255,240,255,255,240
2930 DATA 255,255,236,255,255,204,255
2940 DATA 255,204,255,255,192,255,255
2950 DATA 252,255,255,252,255,255,252
2960 DATA 255,255,255,255,255,255,255
2970 DATA 255,255,255,255,255,255,255
2980 DATA 255,255,255,255,255,255,255
2990 DATA 255,255,255,255,255,255,255
3000 DATA 255,255,255,255,255,255,255:
3010 DATA 255,255,255,192,255,255,207
3020 DATA 255,255,207,255,255,207,255
3030 DATA 255,193,255,255,252,255,255
3040 DATA 252,255,255,204,255,255,193
3050 DATA 255,255,255,255,255,255,255
3060 DATA 255,255,255,255,255,255,255
3070 DATA 255,255,255,255,255,255,255
3080 DATA 255,255,255,255,255,255,255
3090 DATA 255,255,255,255,255,255,255:
3100 DATA 255,255,255,225,255,255,204
3110 DATA 255,255,207,255,255,207,255
3120 DATA 255,193,255,255,204,255,255
3130 DATA 204,255,255,204,255,255,225
3140 DATA 255,255,255,255,255,255,255
3150 DATA 255,255,255,255,255,255,255
3160 DATA 255,255,255,255,255,255,255
3170 DATA 255,255,255,255,255,255,255
3180 DATA 255,255,255,255,255,255,255:
3190 DATA 255,255,255,192,255,255,204
3200 DATA 255,255,252,255,255,252,255
3210 DATA 255,240,255,255,252,255,255
3230 DATA 252,255,255,252,255,255,252
3240 DATA 255,255,255,255,255,255,255
3250 DATA 255,255,255,255,255,255,255
3260 DATA 255,255,255,255,255,255,255
3270 DATA 255,255,255,255,255,255,255
3280 DATA 255,255,255,255,255,255,255:
3290 DATA   0, 24,  0,  0, 60,  0,  0
3300 DATA  60,  0,  0, 60,  0,  0,255
3310 DATA   0,  0,255,  0,  0,255,  0
3320 DATA   0,195,  0,  0,195,  0,  0
3330 DATA 170,  0,  3,170,192, 15, 85
3340 DATA 240, 15, 85,240, 15,170,240
3350 DATA  15,170,240, 15,255,240, 15
3360 DATA 255,240,  0,255,  0,  0,255
3370 DATA   0,  3,195,192,  3,195,192
READY.
```

SORTEO MENSUAL K 64

Se realizó el sorteo mensual que realiza K-64 con los cupones correspondientes que llegan a nuestra redacción. Los premios y los ganadores son los siguientes:

BECAS
Ezequiel Ramos, Néstor F. Raschia, José Luis Varela, Julio Moreno, Juan Jorge Cominetti, Sergio Boghossian, Sergio Jorge Camperi, Esteban Diego Esjaita, Mariano S. Viapiano, Fernando Daniel Medina.

CAMPERA
Pablo Panighini.

CASSETTES
Marcelo Massei, Antonia Peñalver, Manuel C. Vildoza, Mirtha Fabiana Rodríguez, Gustavo Suárez, Néstor Piercamilli, Alejandro Domínguez, Pablo Nicolás Trapani, Marcelo Davicino, Alejandro Rebolledo, Mauricio Mattalia, Mónica B. Núñez, Alejandro Poy, Gabriel Coon, Ernesto Martelli, Miguel A. Ota, Jorge Solanilla, Julio Moreno, Luis Melo, Eduardo Czar, Sebastián Giménez, Oscar Suárez, Diego Navarro, Daniel F. Medina, Carlos Urquiza, Ezequiel Chiappetta, Luis E. Allende, Jorge Parrado, Nidia Cobiella, Luisa Luccione, Gabriela Albornoz, Luis Visciarelli, Ernesto Martelli, Juan M. Vázquez, Fernando Palacio, Héctor Rodofeli, Pablo Grillone, Carlos A. Cejas, Eduardo A. Reynoso, Karina Paladino.

Los premios podrán retirarse en la Administración de K-64, Cerrito 1320, piso 1°, Capital, con documentos de identidad, en el horario: 10 a 12 y 15 a 17 horas. Quienes viven en el interior del país, pueden solicitar que se les remitan los premios por correo.

UTILITARIO DE SONIDO:

COMP: Commodore 64, Disquetera 1540/41
CONF: 64 K
CLAS: UTILITARIO

Este utilitario le permite incorporar tres comandos adicionales al BASIC que serán de gran ayuda al programar distintos sonidos con el chip SID (dispositivo interface de sonido).

Dado que estos comandos están escritos en lenguaje de máquina, el programa BASIC adjunto en su BOOTER o cargador y cumple dos funciones: la primera es generar el código de máquina de las rutinas que componen el utilitario verificando que los valores sean correctos mientras que la segunda consiste en generar un archivo en disco conteniendo sólo el código de máquina del utilitario.

Una vez que el booter en BASIC haya corrido apropiadamente, tipee NEW e inicie su propio programa con las sentencias:

```
0 POKE 52,157: POKE 56,157:CLR
1 IF PEEK(40384)<>76 THEN LOAD "SIDCD",8,1
```

Una vez hecho esto Usted podrá llamar las siguientes tres rutinas:

```
SYS AV,v,a,d,s,r,WC,pw]
SYS PY,v,p,t,vl
SYS QU
```

Donde:

QU=40384 [SYS QU apaga el SID]
PY=QU+3 [SYS PY ejecuta una nota]
AV=QU+6 [SYS AV altera una voz]
v: número de voz (1 a 3)
a: valor de ataque (0 a 15)
d: valor de caída (0 a 15)
s: valor de sostén (0 a 15)
r: valor de relajamiento (0 a 15)
w: tipo de onda:

16- triángulo
32- diente de sierra
64- pulso
128- ruido

pw: ancho de pulso, sólo para onda tipo pulso (0 a 4095)
p: valor del pitch (0 a 86)
t: tiempo en 1/60 segundos de ejecución de la nota
vl: volumen (0 a 15)

Todas las variables descriptas en letras minúsculas pueden ser constantes, variables o expresiones. Por ejemplo, para simular pasos alejándose ingrese la siguiente línea:

```
FOR X=15 TO 0 STEP -1:SYSPY,3,25,30,X:NEXTX
```

```
1 REM ************************************
2 REM *                                  *
3 REM * PROGRAMA UTILITARIO DE SONIDO *
4 REM *                                  *
5 REM *       BOOTER (CARGADOR)          *
6 REM *                                  *
7 REM *  REVISTA K64, JULIO DE 1985.     *
8 REM *                                  *
9 REM ************************************
10 POKE52,157:POKE56,157:POKE51,192:POK
E55,192:CLR:POKE646,PEEK(53280)+1
20 AD=40384:PRINT"(CLR)(C/DN)(C/DN)(C/D
N)(C/DN)(C/DN)(C/DN)(C/RT)(C/RT)(C/RT)(
C/RT)(C/RT)(C/RT)(C/RT)CARGANDO CODIGO
DE MAQUINA."
30 FORL=1TO37:IFL=26THENL=27
40 FORX=0TO15:READB:C=C+B:POKEAD+X,B:NE
XT
50 READK:IFC<>KTHENPRINT"ERROR EN DATA
DE LINEA";1000+L:END
60 T=T+C:C=0:AD=AD+16:NEXTL
70 READK:IFT<>KTHENPRINT"ERROR EN SENTE
NCIAS DATA.":END
80 QU=40384:PY=QU+3:AV=QU+16
90 SYSPY,2,30,30,15:SYSQU
100 PRINT"(CLR)(C/RT)(C/RT)(C/RT)(C/RT)
(C/DN)(C/DN)(C/DN)(C/DN)(C/DN)(C/DN)UTI
LITARIO DE SONIDO CARGADO"
110 PRINT"(C/DN)(C/DN)(C/RT)(C/RT)(C/RT
)(C/RT)GRABANDO EL PROGRAMA EN L.M."
200 F$="SIDCD"
210 POKE187,PEEK(71):POKE188,PEEK(72)
220 FA=PEEK(187)+256*PEEK(188)
230 POKE183,PEEK(FA)
240 POKE187,PEEK(FA+1):POKE188,PEEK(FA+
2)
250 POKE251,192:POKE252,157
260 POKE186,8:POKE185,1
270 POKE780,251:POKE781,0:POKE782,160
280 SYS65496
290 SYSPY,2,50,30,15:SYSQU
300 PRINT"(CLR)(C/DN)(C/DN)(C/DN)(C/DN)
(C/DN)(C/DN)(C/DN)(C/DN)EL UTILITARIO E
N L.M. ESTA GRABADO COMO:(C/DN)(C/RT)(C
/RT)(C/RT)(C/RT)(C/RT)(C/RT)(C/RT)(C/RT
)(C/RT)(C/RT)(C/RT)(C/RT)(C/RT)(C/RT)(C
/RT)"F$
310 PRINT"(C/DN)(C/DN)(C/RT)(C/RT)LA PR
OXIMA VEZ, SOLO TIPEE:"
320 PRINT"(C/DN)(C/DN)(C/RT)(C/RT)LOAD"
CHR$(34)F$CHR$(34)",8,1":POKE646,PEEK(5
3281)
1000 REM
1001 DATA76,45,159,76,78,158,32,69,158,
152,41,3,240,97,141,50,1575
1002 DATA158,206,50,158,32,69,158,152,1
0,10,10,10,133,251,32,69,1508
1003 DATA158,152,41,15,5,251,133,251,32
,69,158,10,10,10,10,133,1438
1004 DATA253,32,69,158,152,41,15,5,253,
133,253,32,69,158,152,41,1816
1005 DATA246,172,50,158,153,57,158,9,1,
153,66,158,165,253,153,54,2006
1006 DATA 158,165,251,153,51,158,185,57
,158,41,64,240,17,32,69,158,1957
1007 DATA72,173,50,158,10,170,104,157,6
1,158,152,157,60,158,96,100,1844
1008 DATA0,3,0,9,10,9,3,3,9,64,64,128,0
,4,0,4,310
1009 DATA0,0,65,65,129,32,253,174,32,15
8,173,76,170,177,32,69,1605
1010 DATA158,152,41,3,240,33,141,50,158
,206,50,158,32,69,158,132,1781
1011 DATA251,32,69,158,132,253,32,69,15
8,140,24,212,173,50,158,240,2151
1012 DATA11,74,176,62,76,232,158,162,14
,108,0,3,173,51,158,141,1599
1013 DATA5,212,173,54,158,141,6,212,173
,60,158,141,2,212,173,61,1941
1014 DATA158,141,3,212,166,251,189,167,
159,188,80,159,141,0,212,140,2366
1015 DATA1,212,173,66,158,141,4,212,32,
30,159,173,57,158,141,4,1721
1016 DATA212,96,173,52,158,172,55,158,1
41,12,212,140,13,212,173,62,2041
1017 DATA158,141,9,212,173,63,158,141,1
0,212,166,251,189,167,159,188,2397
1018 DATA80,159,141,7,212,140,8,212,173
,67,158,141,11,212,32,30,1703
1019 DATA159,173,58,158,141,11,212,96,1
73,53,158,172,56,158,141,19,1930
1020 DATA212,140,20,212,173,64,158,141,
16,212,173,65,158,141,17,212,2114
1021 DATA166,251,189,167,159,188,80,159
,141,14,212,140,15,212,173,68,2334
1022 DATA158,141,18,212,32,30,159,173,5
9,158,141,18,212,96,160,0,1767
1023 DATA162,10,136,208,253,202,208,250
,198,253,208,244,96,169,0,141,2738
1024 DATA24,212,141,4,212,141,11,212,14
1,18,212,96,0,141,24,212,1801
1025 DATA141,4,212,141,11,212,141,18,21
2,96,255,255,255,255,255,255,2718
1026 REM
1027 DATA1,1,2,2,2,2,2,2,2,3,3,3,3,3,4,
4,39
1028 DATA4,4,5,5,5,6,6,6,7,7,8,8,9,9,10
,10,109
1029 DATA11,12,12,13,14,15,16,17,18,19,
20,21,22,24,25,27,286
1030 DATA28,30,32,34,36,38,40,43,45,48,
51,54,57,61,64,68,729
1031 DATA72,76,81,86,91,96,102,108,115,
122,129,137,145,153,163,172,1848
1032 DATA183,193,205,217,230,244,0,205,
233,6,37,69,104,140,179,220,2465
1033 DATA8,54,103,155,210,12,73,139,208
,25,103,185,16,108,206,53,1658
1034 DATA163,23,147,21,159,50,205,114,3
2,216,156,107,70,47,37,42,1589
1035 DATA63,100,154,227,63,177,56,214,1
41,94,75,85,126,200,52,198,2025
1036 DATA127,97,111,172,126,188,149,169
,252,161,105,140,254,194,223,88,2556
1037 DATA52,120,43,83,247,31,210,25,252
,133,109,176,103,0,0,0,1664,62717
9999 STOP
10000 OPEN15,8,15,"S:SOUND.LDR":CLOSE15
:SAVE"SOUND.LDR",8:VERIFY"SOUND.LDR",8

READY.
```

GLOSARIO DE TERMINOS INFORMATICOS

LETRA "D"

D/A - DIGITAL TO ANALOG:
La conversión o el dispositivo que transforma información digital en señal analógica, como por ejemplo la conversión del número binario 11001 en un voltaje de 25 Volts.

DATA REGISTERS:
Registros o memorias especiales dentro del CPU, destinados al almacenamiento temporario de información.

DEBOUNCING:
Dispositivo circuital que evita los problemas de microrebote generado en los teclados de computadoras al pulsar una tecla.

DEBUGGING:
Proceso de depuración y corrección de un programa. Típicamente consiste en pruebas de verificación y chequeo de validez de los algoritmos, en busca de errores.

DIGITAL MAGNETIC TAPE STORAGE:
Sistema de almacenamiento de datos que usa la tecnología de las cintas magnéticas. Cuando se reproduce la cinta, los segmentos magnetizados producen pulsos electrónicos que representan valores binarios.

DISK CRASH:
Se refiere a la falla presentada en un sistema de almacenamiento en disco.

DISKETTE:
Disco plástico recubierto de material magnético, usado como soporte permanente de información, tanto de datos como de programas, etc. Suelen fabricarse en distintos formatos y tamaños.

DISPLAY:
Todo elemento electrónico destinado a mostrar información. Puede ser una pantalla o dígitos como en una calculadora.

DOT MATRIX CRT DISPLAY:
Pantalla que opera iluminando puntos seleccionados en una red de puntos.

DOS - DISK OPERATING SYSTEM:
Programa que se encarga del control de todas las operaciones con el disco. Entre los más difundidos están el CP/M y el MSDOS.

DOWNLOAD:
Transferencia de información entre ordenadores.

DRIVE:
La unidad de control de disco.

DYNAMIC MEMORY ELEMENTS:
Memoria de semiconductores que no permite el almacenamiento de información por más de unas pocas milésimas de segundo, a menos que un circuito especial le "refresque" continuamente esos datos.

PARA DUPLICAR SOFTWARE

Ing. Pedro E. Colla

Debido al efecto negativo que el primer caso representa sobre los que se dedican a la creación de software, es que el mismo viene provisto de "trabas" que tienden a dificultad, sino impedir, su duplicación.

Esta modalidad, empero, tiende a dificultad o impedir la generación de copias ante la eventualidad que los medios que contienen los programas, normalmente cassettes, resulten dañados con el consiguiente trastorno asociado.

El Presente, íntegramente confeccionado en lenguaje de máquina, permite realizar una lectura desde cassette almacenando en memoria el programa o bloque de datos para a continuación volcarlo nuevamente a cassette, permitiendo de esa manera la realización de duplicados.

El programa, cuyo listado aparece en la Figura 1, ocupa en su versión ya ensamblada alrededor de 600 bytes de memoria permitiendo almacenar en memoria segmentos cuya longitud sea de hasta 34 K, más que suficiente para la mayoría de las aplicaciones.

El ingreso del mismo se debe llevar a cabo con el auxilio de un Programa Ensamblador, siendo en este caso utilizado el denominado ZEUS-ASSEMBLER para TS2068, aunque no es difícil su adaptación para utilizar cualquier otro.

Cabe destacar, que debido al intensivo uso de rutinas del ROM del computador el mismo **no** funcionará en máquinas SPECTRUM por diferencias en las direcciones de tales rutinas.

El programa comienza definiendo los mensajes denominados MS1, MS2,MS3,MS4,MS5 y MS6, los cuales son textos que se utilizarán en distintas etapas para orientar al usuario en los pasos a seguir, así como para exhibir títulos, mensajes, etc.

A continuación se define un área de memoria destinada a albergar el **header** del programa a copiar, tal como reside en la cinta de audio.

Al respecto, es válido indicar brevemente el formato en el cual se graban los programas en este computador;

Cuando se da la orden de comenzar la grabación la rutina encargada de esta tarea emite un período de **sincronismo** compuesto por 4032 ciclos a una frecuencia de 806.5 Hz, esto tiene el propósito de permitir que posteriormente la rutina de lectura se "enganche" con los datos que vienen a continuación.

Continúa al sincronismo un bloque de 17 bytes de longitud el cual recibe la denominación de **header** que contiene la siguiente información:

- Tipo de Datos (1 byte)
  00 Programa BASIC
  01 Vector/Matriz Numérica
  02 Vector/Matriz Alfanumérica
  03 Bloque de Memoria (Bytes)
- Nombre del Bloque (10 bytes)
- Longitud del Bloque en Bytes (2 bytes)
- Dirección/Línea de Comienzo (2 bytes)
- Datos auxiliares (2 bytes)

Al finalizar el bloque se encuentra un código (CRC) con el cual el computador puede reconocer si la información que ha leído es correcta, este código está formado mediante la realización de la operación OR-Exclusiva entre todos los formantes del bloque.

Luego del header, se produce un retardo de 835 mSeg y se graba el bloque de programa/datos que corresponda, en este caso el sincronismo tiene la misma función pero sólo se prolonga durante 1612 ci-



La duplicación de software puede ser utilizada básicamente con dos fines, uno de ellos ilegítimo, la copia de programas para evitar su adquisición, y el segundo no sólo legítimo sino también necesario, la creación de duplicados de seguridad.

clos.
Este formato es válido para computadores TS2068 y SPECTRUM únicamente.
El Timex-SINCLAIR 2068 posee dos integrados de ROM, a diferencia del SPECTRUM que solamente posee uno. El primero de ellos, de 16 K de longitud, contiene las rutinas de teclado, video normal, sonido, el intérprete BASIC y lo que recibe el nombre de **calculador de punto flotante** utilizado tanto por el BASIC como por otras secciones del ROM. A diferencia del SPECTRUM existre otro segmento de ROM, de 8 K de longitud, denominado **ROM extendido** el cual contiene las rutinas de LOAD/SAVE y ciertas rutinas especiales para el manejo de los modos de video que dispone el TS 2068.
La particularidad es que ambos, que suman 24 K, solamente ocupan 16 K del mapa de memoria, ésto es están superpuestos.
A su vez, cualquier "cartridge" que se coloque en el "buzón" delantero del computador, compartirá el espacio con el ROM principal. Por lo tanto cada vez que se ejecuta una rutina que está dentro del ROM extendido o del cartridge es necesario realizar previamente un proceso de habilitación especial, debiendo obligatoriamente realizar el paso inverso cuando se requiere continuar procesando con el ROM normal.
Todas las operaciones de Entrada/Salida que realiza el TS2068 (exceptuando lectura/grabación en cassette) se manejan a través de abstracciones de software denominadas **canales**, esto es dichas operaciones son realizadas con independencia de cuál es el elemento de E/S a ser utilizado, accediéndose a una tabla residente en memoria RAM para determinar la rutina de bajo nivel que en definitiva realizará la operación de lectura/grabación concreta. Esta metodología puede lucir confusa e ineficiente, pero totalmente al contrario es extremadamente poderosa, pues permite que un dispositivo no standard, originalmente no previsto en el ROM, pueda ser adicionado con posterioridad mediante el simple expediente de **abrir** un canal (con OPEN #), asignarlo al dispositivo y asegurarse que en la ta-

bla respectiva se indica claramente donde debe ir el computador a buscar la "rutina" que le permite utilizar ese dispositivo.

En el ROM original solamente existen especificaciones para el teclado, la pantalla de video normal (22 líneas superiores), la impresora térmica y la pantalla de video del sistema (líneas inferiores) siendo sus números standard de canal los siguientes:

Canal 0 - Parte baja de la pantalla
Canal 1 - Teclado
Canal 2 - Pantalla Normal
Canal 3 - Impresor térmico

Se permiten hasta 15 canales diferentes, debiendo en todos los casos agregar cuando no se trate de un dispositivo standard la rutina que realice el manejo del mismo, pues como es lógico no figura en el ROM.

El programa comienza indicando que las salidas las realizará a través del canal # 2, esto se lleva a cabo mediante el llamado a la rutina del ROM normal que comienza en #1230. A continuación se borra la pantalla recurriendo nuevamente a otra rutina contenida en dicha sección (#08EA).

Se coloca el cursor en la posición 0,0 (extremo superior izquierdo) mediante la llamada a la rutina que comienza en la dirección #0582 con el acumulador B conteniendo el número de línea y el C el número de columna.

A continuación, se exhiben dos de los mensajes de programa, lo que se realiza con la rutina MSSG, la cual toma la cantidad indicada de caracteres y los envía a través de la sección del ROM para salida de datos normalizada, de allí que se seleccionara previamente el canal 2, esto es la pantalla.

Las rutinas LHDR, LPRG, SHDR y SPRG son muy similares entre sí y básicamente colocan en el registro HL la dirección dentro del ROM extendido de la rutina que se utilizara (LOAD o SAVE según el caso), en el registro índice IX se coloca el comienzo del área de memoria de donde se tomarán/sacarán los datos (HDR o 30600 dependiendo de si es un Header o datos) y en el registro DE la longitud de los mismos. Nótese que cuando se lee o graba los datos en sí la información de longitud es extraída del área de Header.

La rutina IFRT realiza la conmutación entre ROM normal y ROM extendido, siendo el método expuesto standard para ello; mientras que la rutina EXIT realiza el procedimiento inverso.

La secuencia de ejecución es leer el header del programa, imprimir el nombre, tipo y longitud del mismo y a continuación leer el programa en sí mismo a partir de la dirección 30600.

Una vez finalizada la carga, se espera la indicación adecuada y se vuelcan nuevamente tanto header como datos/programa tantas veces como se desee.

Para finalizar la ejecución bastará tener oprimida la tecla de break.

En la rutina que exhibe las características del programa a duplicar se tiene en cuenta el formato y posibles contenidos del header del mismo, exhibiéndose mensajes apropiados. Nótese que se utiliza una rutina de conversión del ROM para trasladar la longitud expresada en binario a caracteres ASCII.

En la rutina de retardo (DLY) simplemente se "pierde tiempo" haciendo contar al computador hasta 65500, ésto no obstante tarda solamente algunos miles de milisegundos! Como técnica de programación debe apuntarse que no se finaliza el loop en la forma normal (RET Z o JR Z) luego de decrementar BC, pues esta operación NO modifica el FLAG Z (flag que indica que la última operación matemática dio resultado 0).

En la rutina de espera (WRIT) simplemente se chequea el teclado hasta que se detecta que hay cualquier tecla oprimida. Para hacer esta misma tarea se podría utilizar el método formal, es decir abrir el canal 1, leerlo y determinar si hay o no información proveniente del mismo. Esto es molesto por la necesidad de abrir el canal 1 para luego al finalizar, abrir nuevamente el canal 2 para cualquier operación de escritura sobre la pantalla. En cambio se aprovecha el hecho de que, independientemente de lo que hagamos, el computador "lee" cada 16.5 mSeg el teclado y coloca el caracter correspondiente a la tecla oprimida en la posición 23560 (LAST-K), por lo que bastará colocar 0 en ésta y luego darle tiempo para que por sí sólo se encargue de colocar allí algún valor, si lo hubiere.

En el caso típico de que el programa a copiar esté compuesto de varios segmentos, se deberá realizar la duplicación de cada segmento en forma independiente.

TS2068

```
L
ORC 30000
;-------------------------------
;COPY
;PROGRAMA PARA COPIAR
;SOFTWARE EN CASSETTE
;-------------------------------
PEC JP COPY
;-------------------------------
;AREA DE MENSAJES
;-------------------------------
MS1 DEFM / COPIADOR V1.0/
    DEFB 13,13
MS2 DEFM /Colocar Progr/
    DEFM /ama en reprod/
    DEFM /uctor/
MS3 DEFB 13,13
    DEFM /Colocar casse/
    DEFM /tte virgen/
    DEFB 13
    DEFM /Luego oprimir/
    DEFM / cualquier te/
    DEFM /cla /
    DEFB 13
    DEFM /Para grabar/
MS4 DEFM /ERROR! Pulsar/
    DEFM / cualquier te/
    DEFM /cla          /
MS5 DEFM / /
MS6 DEFM / Bytes /
;-------------------------------
;BUFFER DONDE SE ALMACE-
;NA EL HEADER DEL PRO-
;GRAMA A REPRODUCIR.
;   HDR - 00 bytes
;         01 Program
;         02 num array
;         03 char array
;   NAM - NOMBRE
;   LEN - LARGO
;   ADR - ADRESS START
;-------------------------------
HDR DEFB #00
NAM DEFM /          /
LEN DEFB #00,#00
ADR DEFB #00,#00
CTR DEFB #00,#00,13,13
;-------------------------------
;TABLA DE TIPOS DE DATOS
;POSIBLES DE ENCONTRAR
;-------------------------------
TPR DEFM / Prog /
BPR DEFM / byte /
APR DEFM / data /
EPR DEFM / ERROR/
;-------------------------------
;COMIENZA EL PROGRAMA.
;SELECCIONA DISPOSITIVO
;DE SALIDA #2 (video).
;-------------------------------
COPY LD A,2
     CALL #1230
;-------------------------------
;BORRA LA PANTALLA (cls)
;-------------------------------
     CALL #08EA
;-------------------------------
;COLOCA EL CURSOR EN 0.0
;-------------------------------
     LD BC,#0000
     CALL #0582
;-------------------------------
;EXHIBE NOMBRE DE PROGR.
;-------------------------------
     LD HL,MS1
     LD B,16
     CALL MSSG
;-------------------------------
;PIDE COLOCAR CASSETTE
;-------------------------------
     LD HL,MS2
     LD B,33
     CALL MSSG
;-------------------------------
;LEE EL HEADER DEL PROGR
;-------------------------------
```



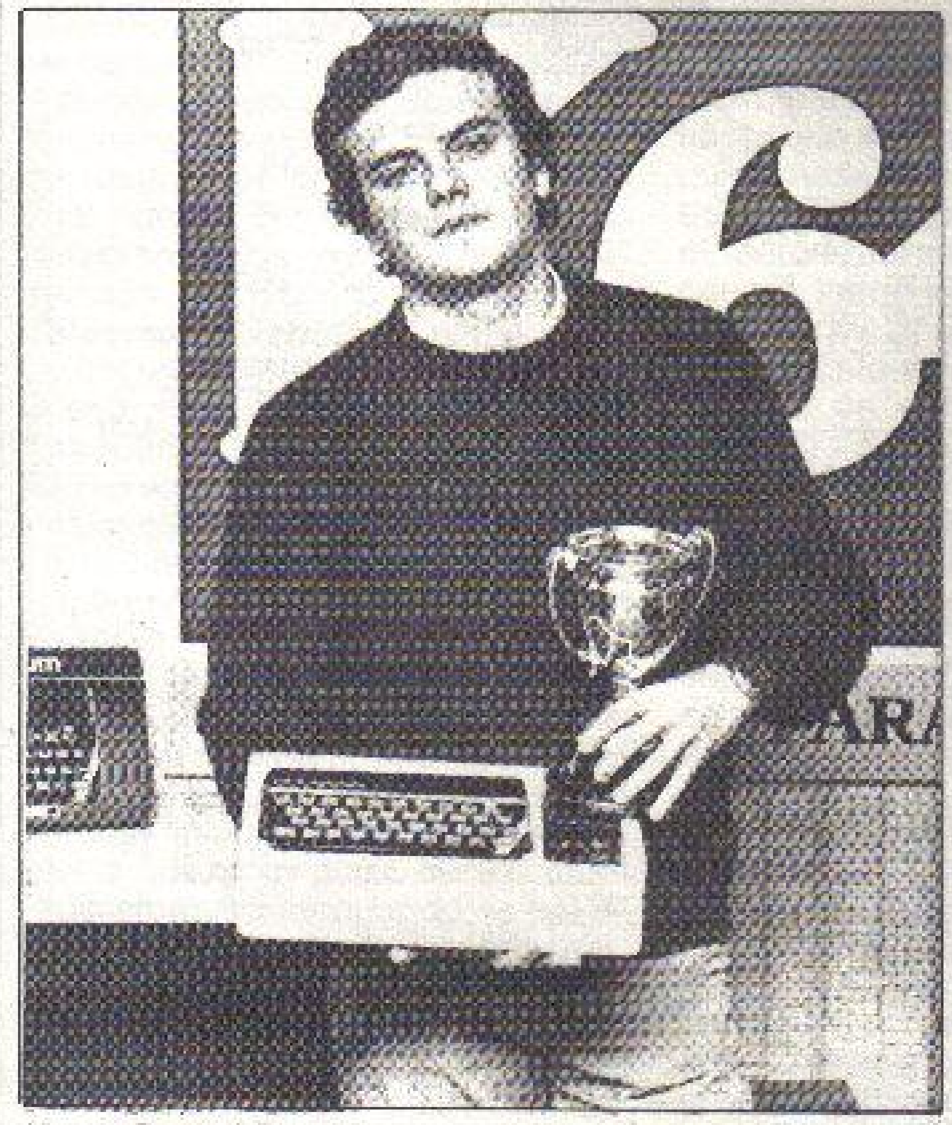

```
00860       CALL LHDR
00870 ;------------------------
00880 ;SI DETECTA ERROR VA A
00890 ;RUTINA 'BAD'
00900 ;------------------------
00910       JR NC,BAD
00920 ;------------------------
00930 ;EXHIBE NOMBRE DEL PROGR
00940 ;LEIDO Y LOS DATOS DEL
00950 ;MISMO
00960 ;------------------------
00970       CALL DHDR
00980 ;------------------------
00990 ;LEE EL PROGRAMA
01000 ;------------------------
01010       CALL LPRG
01020 ;------------------------
01030 ;RETARDO POR SOFTWARE
01040 ;------------------------
01050       CALL DLY
01060 ;------------------------
01070 ;AL TERMINAR DE CARGAR
01080 ;INVITA AL CAMBIO DE
01090 ;CASSETTE
01100 ;------------------------
01110 SPR  LD HL,MS3
01120       LD B,68
01130       CALL MSSG
01140 ;------------------------
01150 ;ESPERA HASTA QUE SE
01160 ;OPRIMA CUALQUIER TECLA
01170 ;------------------------
01180       CALL WAIT
01190 ;------------------------
01200 ;SALVA EL HEADER TAL
01210 ;COMO LO LEYO
01220 ;------------------------
01230       CALL SHDR
01240 ;------------------------
01250 ;INTRODUCE UN RETARDO
01260 ;------------------------
01270       CALL DLY
01280 ;------------------------
01290 ;SALVA EL RESTO DEL PROG
01300 ;------------------------
01310       CALL SPRG
01320 ;------------------------
01330 ;VA A SALVAR NUEVAMENTE
01340 ;------------------------
01350       JR SPR
01360 ;------------------------
01370 ;RUTINA DE ERROR DE
01380 ;LECTURA
01390 ;POSICIONA EL CURSOR EN
01400 ;lin 21 col 0 Y EXHIBE
01410 ;MENSAJE DE ERROR
01420 ;------------------------
01430 BAD  LD B,21
01440       LD C,0
01450       CALL #05B2
01460       LD HL,MS4
01470       LD B,32
01480       CALL MSSG
01490 ;------------------------
01500 ;ESPERA QUE SE OPRIMA
01510 ;CUALQUIER TECLA
01520 ;------------------------
01530       CALL WAIT
01540 ;------------------------
01550 ;RETORNA AL COMIENZO
01560 ;------------------------
01570       JR COPY
01580 ;------------------------
01590 ;RUTINA DE LECTURA DE
01600 ;HEADER. SE CARGAN RE-
01610 ;GISTROS CON:
01620 ;HL-Direccion de ROM
01630 ;   de la rutina de LOAD
01640 ;IX-Direccion donde vol-
01650 ;   cara los datos que
01660 ;   lea.
01670 ;DE-Longitud del bloque a
01680 ;A -Con #00 indica header
01690 ;------------------------
01700 LHDR LD HL,#00FC
01710       LD IX,HDR
01720       LD A,#00
01730       SCF
01740       LD DE,17
01750       CALL IFRT
01760       JR EXIT
01770 ;------------------------
01780 ;RUTINA DE LECTURA DEL
01790 ;CUERPO DEL PROGRAMA
01800 ;idem LHDR excepto que
01810 ;IX apunta ahora al un
01820 ;punto de memoria RAM.
01830 ;DE se saca del HEADER
01840 ;A  es #FF indicando prog
01850 ;------------------------
01860 LPRG LD HL,#00FC
01870       LD A,#FF
01880       LD IX,30600
01890       SCF
01900       LD DE,(LEN)
01910       CALL IFRT
01920       JR EXIT
01930 ;------------------------
01940 ;SALVADO DEL HEADER
01950 ;idem anterior excepto
01960 ;que HL apunta ahora a la
01970 ;rutina de SAVE
01980 ;------------------------
01990 SHDR LD HL,#0068
02000       LD A,#00
02010       LD IX,HDR
02020       LD DE,17
02030       CALL IFRT
02040       JR EXIT
02050 ;------------------------
02060 ;SALVADO DEL PROGRAMA
02070 ;idem anteriores
02080 ;------------------------
02090 SPRG LD HL,#0068
02100       LD A,#FF
02110       LD IX,30600
02120       LD DE,(LEN)
02130       CALL IFRT
02132       JR EXIT
02140 ;------------------------
02150 ;RUTINA DE HABILITACION
02160 ;DEL ROM DE EXTENSION
02170 ;------------------------
02180 IFRT DI
02190       PUSH AF
02200       IN A,(#FF)
02210       SET 7,A
02220       OUT (#FF),A
02230       IN A,(#F4)
02240       LD (HSAV),A
02250       LD A,1
02260       OUT (#F4),A
02270       POP AF
02280 ;------------------------
02290 ;SALTA A LA RUTINA CUYA
02300 ;DIRECCION ESTA CARGADA
02310 ;EN EL REGISTRO HL, EL
02320 ;CUAL TENDRA UN VALOR
02330 ;DEPENDIENTE DE LA RUTI-
02340 ;NA DE LLAMADA.
02350       JP (HL)
02360 ;------------------------
02370 HSAV DEFB #00
02380 ;------------------------
02390 ;RESTAURA EL ROM NORMAL
02400 ;DEL PROCESADOR
02410 ;------------------------
02420 EXIT LD A,(HSAV)
02430       OUT (#F4),A
02440       IN A,(#FF)
02450       RES 7,A
02460       OUT (#FF),A
02470       EI
02480       RET
02490 ;------------------------
02500 ;EXHIBE LOS DATOS DEL
02510 ;PROGRAMA TAL COMO FIGURA
02520 ;EN EL HEADER.
02530 ;PRIMERO DETECTA EL TIPO
02540 ;DE PROGRAMA.
02550 ;------------------------
02560 DHDR LD A,(HDR)
02570       CP #00
02580       JR Z,PROG
02590       CP #01
02600       JR Z,NARR
02610       CP #02
02620       JR Z,CARR
02630       CP #03
02640       JR Z,BYTE
02650       LD HL,EPR
02660       JR EXH
02670 PROG LD HL,TPR
02680       LD B,6
02690       JR EXH
02700 NARR LD HL,APR
02710       JR EXH
02720 CARR LD HL,APR
02730       JR EXH
02740 BYTE LD HL,BPR
02750       JR EXH
02760 ;------------------------
02770 ;IMPRIME TIPO DETECTADO
02780 ;------------------------
02790 EXH  LD B,6
02800       CALL MSSG
02810 ;------------------------
02820 ;IMPRIME NOMBRE
02830 ;------------------------
02840       LD B,10
02850       LD HL,NAM
02860       CALL MSSG
02870       LD HL,MS5
02880       LD B,1
02890       CALL MSSG
02900 ;------------------------
02910 ;CONVIERTE LARGO DEL
02920 ;PROGRAMA DE BINARIO A
02930 ;CARACTERES ASCII Y LOS
02940 ;IMPRIME, PREVIA IMPRE-
02950 ;SION DE UN CARACTER
02960 ;ESPACIO COMO SEPARADOR
02970 ;LUEGO COLOCA EL TEXTO
02980 ;FIJO 'bytes'.
02981 ;------------------------
02990       LD BC,(LEN)
03000       CALL #1783
03010       LD B,7
03020       LD HL,MS6
03030       CALL MSSG
03040       RET
03041 ;------------------------
03042 ;RETARDO POR SOFTWARE
03043 ;------------------------
03050 DLY  PUSH BC
03060       LD BC,65500
03070 DLY1 DEC BC
03080       LD A,B
03090       OR C
03100       JR NZ,DLY1
03110       POP BC
03120       RET
03121 ;------------------------
03122 ;EXHIBE MENSAJES EN
03123 ;PANTALLA.
03124 ;EN EL REGISTRO HL DEBERA
03125 ;FIGURAR LA DIRECCION DE
03126 ;COMIENZO DEL MENSAJE Y
03127 ;EN EL REGISTRO B SU LON-
03128 ;GITUD.
03129 ;------------------------
03130 MSSG LD A,(HL)
03140       PUSH BC
03150       PUSH HL
03160       CALL #0500
03170       POP HL
03180       POP BC
03190       INC HL
03200       DJNZ MSSG
03210       RET
03220 ;------------------------
03221 ;ESTA RUTINA ESPERA QUE
03222 ;SE OPRIMA CUALQUIER TE-
03223 ;CLA Y LUEGO RETORNA
03224 ;------------------------
03230 WAIT XOR A
03240       LD (23560),A
03250       CALL DLY
03260       LD A,(23560)
03270       CP #00
03280       JR Z,WAIT
03290       RET
34000 FIN  NOP
```

GANADORES DEL

Por fin llegó la evaluación y resultado del primer gran concurso de K-64. El trabajo de revisión y análisis del material recibido fue realmente arduo, ya que gran parte del mismo es de gran calidad. Finalmente hubo que decidirse sólo por cinco finalistas, pero seguramente deberían haber sido más. La variedad de temas recibidos, sobre los cuales estaban diseñados los programas, hizo aún más difícil la tarea de selección.
Gracias por participar, y a aquéllos que no les tocó esta vez, no crean que estuvieron muy lejos; inténtenlo nuevamente!
Debemos destacar aquí la invalorable ayuda prestada en la selección, por distintos colaboradores y profesionales, entre quienes destacamos a Patricia Micelli y Agueda Torres.
Como indicamos previamente, el criterio se basa en la evaluación de varias características y cualidades del programa, entre las cuales figuran la originalidad de la idea; método y calidad de programación; gráficos y/o sonido utilizados, y documentación presentada en el programa o aparte. Luego de la evaluación detallada de cada ítem por nuestros colaboradores, se realizó la elección.
Los ganadores son:
Primer Premio: Una computadora CZ 2000.
Nombre: HUGO D. BUSSO (TOLOSA).
Programa. Gráficos comerciales (publicado en el Nº 3 de K 64)
Computadora: CZ2000/TS2068/TK90X
Configuración: 16 K
Clasificación: Comercial.
Segundo Premio: Una mesa Epicom para cualquier tipo de microcomputadores y sus accesorios.
Nombre: CARLOS H. MARINA (BS. AS).
Programa: Pesca (se publica en las siguientes páginas).
Computadora: TS2068.
Configuración: Joysticks.
Clasificación: Entretenimiento.
Mención: E.L. VARETTI (LA PLATA)
Programa: Pant (se publica en esta edición).
Computadora: CZ1000/1500 TK 83/85.
Configuración: 16 K.
Clasificación: Utilitario.
Mención: MANUEL A. LOPEZ
Programa: Invasión de los seteros (lo incluiremos en el próximo número de K64).
Computadora: CZ1000/1500 TK 83/85
Configuración: 16 K.
Clasificación: Entretenimiento.
Mención: MARCELO Y JULIAN VALOTA (BS. AS.).
Programa: Ruleta Rusa (se publicará en la próxima edición de K64).
Configuración: 16 K.
Computadora: CZ1000/1500 TK 83/85.
Clasificación: Entretenimiento.

¿QUIENES SON LOS PREMIADOS?

Hugo Daniel Busso (25 años, casado hace 6 meses) es estudiante de Agrimensura y de Ingeniería en Construcciones; cursa el quinto año en la Universidad de La Plata. Empezó a aprender computación por sus propios medios, hace tres años, con un manual y con una Sinclair 1000 de las que recién llegaban al país. "Con libros y sobre todo con muchas ganas llegué a dominar el tema", nos dijo. Hace un par de años comenzó a hacer programas para resolver problemas de la facultad o para trabajos prácticos.
Para el concurso, comentó, tenía cuatro posibilidades: hacer un programa educativo, comercial, de jue-



Hugo Daniel Busso.



Carlos Hernán Marina



gos de acción o estratégico. La alternativa de juegos de acción le hubiera demandado hacer "un programa en assembler, que hubiera sido muy largo y no lo dominaba muy bien, y hubiera requerido más tiempo para lograr un trabajo óptimo". Le gusta el tema de la educación para chicos, "pero un programa de esta naturaleza necesito mucha chispa e imaginación". Por eso las descartó. Ante las otras opciones, decidió —por gusto personal— la vía elegida y premiada.
Trabajó en una Spectrum prestada, alrededor de dos semanas, dedicándole varias horas diarias luego del estudio y las clases, "gracias al apoyo y la comprensión de mi esposa, ya que usaba el único televisor que tenemos y llevaba la casa deapuntes y papeles". No sabemos si ella estará contenta, porque ahora Hugo tiene su Spectrum, que usará mucho tiempo. Le aconsejamos ganar un concurso donde los premien con un televisor.
Carlos Marina, 19 años, cursa el ciclo básico común para estudiar Análisis de Sistemas en la Facultad de Ingeniería de la Universidad Nacional de Buenos Aires. Tampoco hizo ningún curso. Se compró una Timex Sinclair hace un año. Primero hizo programas para los prácticos del colegio (ENET Nº 1, San Ginés de San Fernando). Y también se lanzó a hacer juegos. Preparó uno parecido al de Pesca que obtuvo el segundo premio. Cuando se enteró que K 64 hacía un concurso, lo perfeccionó y lo presentó.
¿Cómo se le ocurrió este tema? "Me gusta la pesca, y la practico con frecuencia. Hice el programa en varias etapas; primero había un solo pescador, y luego le fui agregando cosas".
Contrariamente a lo que le sucede a Hugo, Carlos sí tiene un segundo televisor para usar exclusivamente con la computadora, pero le faltaba la mesa, que ahora —gracias a su inteligencia y perseverancia— ya posee al haberse adjudicado el segundo premio del Concurso K 64.



PANT.

COMP: CZ1000/1500, TK83/85
1500, TK83/85
CONF: 16 K
CLAS: UTILITARIO

Autor: E.L. Varetti

Se presenta un programa utilitario destinado a generar, almacenar y reproducir una serie de figuras o "pantallas". Se desarrolló este programa con fines educativos, a fin de presentar fenómenos de carácter cíclico en forma animada en la pantalla del televisor. A fin de lograr suficiente velocidad de operación, los procesos de almacenamiento (por encima de RAMTOP) y recuperación de imágenes se realiza por medio de breves programas en lenguaje de máquina ubicados en la línea 1.
Aunque existen programas comerciales con objetivos similares a los aquí descritos, los mismos son, a juicio del autor, demasiado largos y complejos. Este hecho hace que tome considerable tiempo cargarlos desde el casette y sea dificultoso "meter mano" en ellos para adaptarlos a diferentes necesidades.
En el casette se han grabado dos aplicaciones del programa, que se describen a continuación:

CONCURSO



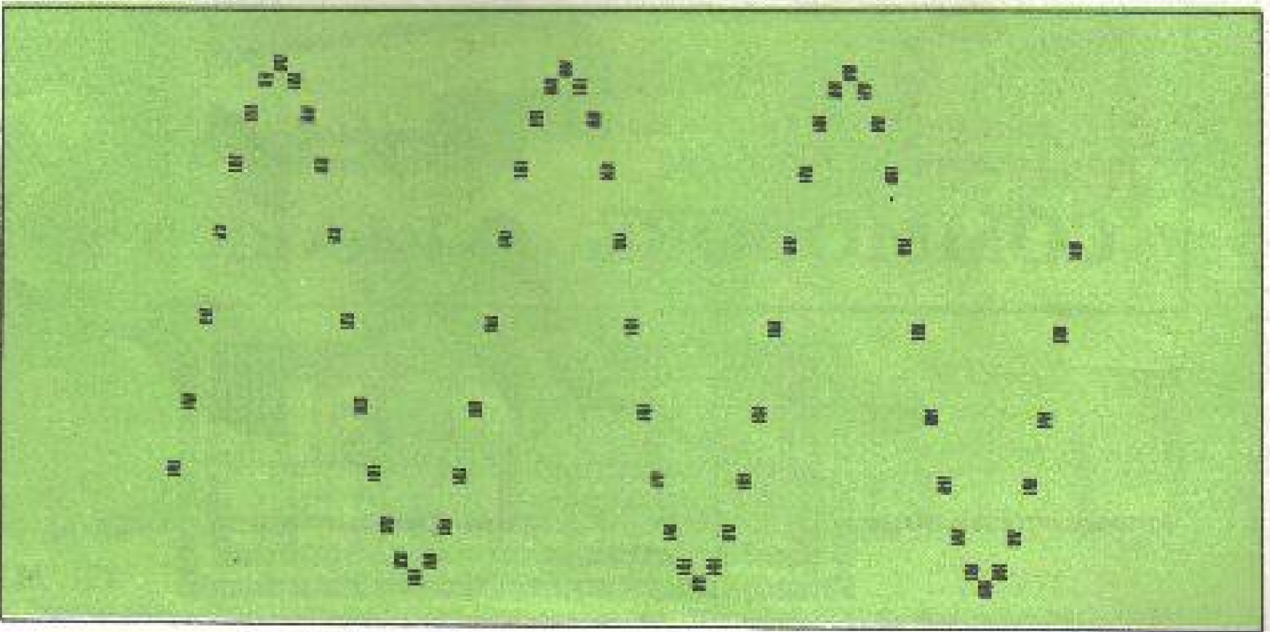
PANT.SEN: Puesto en marcha el programa con RUN, debe contestarse 10 veces con "S", "ENTER" a la pregunta "IMAGEN 1?" (S/N)". A partir de la línea 700 se ubica la subrutina de generación de las imágenes, que dependerá de cada aplicación. En este caso, se generan curvas senoidales desplazadas entre sí. Luego de esta primera parte debe contestarse a la pregunta "QUE FIGURA ?" con los números de las figuras antes generadas, en el orden que se quiera. En este caso, se presentan todas las figuras en el orden original; entonces se contesta "12345 6789A". Se contesta luego a la pregunta "VELOCIDAD ? (1-100)" (a mayor valor menor velocidad), apareciendo a continuación las figuras en pantalla. Pulsar "F" (fin) para cambiar figuras y/o velocidad.

LISTADO EN LENGUAJE DE MAQUINA

16514=30	16515=59
16516=64	16517=237
16518=75	16519=120
16520=64	16521=203
16522=33	16523=9
16524=94	16525=35
16526=86	16527=235
16528=237	16529=91
16530=12	16531=64
16532=1	16533=25
16534=3	16535=237
16536=176	16537=201
16538=42	16539=12
16540=64	16541=237
16542=91	16543=123
16544=64	16545=1
16546=25	16547=3
16548=237	16549=176
16550=201	
16552=0	

```
   1 REM 5▒RND GOSUB ??RNDACS 5▒
??? FOR GOSUB ?£RND";▀ GOSUB ▒TA
N E£RND GOSUB ??RND";▀ GOSUB ▒TA
N
   3 REM PROGRAMA PANT. EJEMPLO
DE APLICACION CON CURVAS SENOIDA
LES. E.L.VARETTI, LA PLATA. 1985
.
 100 POKE 16388,0
 110 POKE 16389,94
 120 LET T=PEEK 16388+256*PEEK 1
6389
 121 FOR I=1 TO 10
 123 LET P=T+(I-1)*800
 130 POKE (16443+2*I),P-256*INT
(P/256)
 140 POKE (16444+2*I),INT (P/256
)
 150 NEXT I
 160 CLS
 161 REM ***UTILIZAR LINEAS 161
A 169 PARA DEFINIR Y/O DIMENSION
AR VARIABLES NECESARIAS PARA GEN
ERAR LAS FIGURAS.
 165 DIM C(64)
 170 LET A=1
 175 REM ***GENERACION Y ALMACEN
AMIENTO DE IMAGENES (10 COMO MAX
IMO).
 180 PRINT "IMAGEN ";A;"?..(S/N
"
 190 INPUT B$
 200 IF B$="N" THEN GOTO 400
 210 GOSUB 700
 280 POKE 16507,PEEK (16443+2*A)
 290 POKE 16508,PEEK (16444+2*A)
 300 RAND USR 16538
 310 LET A=A+1
 320 IF A<=10 THEN GOTO 180
 400 CLS
 405 SLOW
 407 REM ***PRESENTACION SECUENC
IAL DE LAS IMAGENES PREVIAMENTE
GENERADAS.
 410 PRINT "QUE FIGURA?"
 420 INPUT A$
 421 PRINT "VELOCIDAD? (1-100)"
 422 INPUT B$
 423 LET B=VAL B$
 425 FOR I=1 TO LEN A$
 427 FOR J=1 TO B
 428 NEXT J
 440 POKE 16507,CODE A$(I)-28
 445 POKE 16508,0
 450 RAND USR 16511
 455 NEXT I
 457 IF CODE INKEY$=43 THEN GOTO
410
 460 GOTO 425
 590 REM ***SUBRUTINA APROPIADA
PARA GENERAR LAS IMAGENES. EN ES
TE CASO SE GENERAN CURVAS SENOID
ALES DESPLAZADAS ENTRE SI.
 700 CLS
 715 LET B=A*2
 720 FOR V=1 TO 64
 730 LET C(V)=22+15*(SIN ((V-B)/
10*PI))
 740 PLOT V-1,C(V)
 750 NEXT V
 770 RETURN
```

PANTALLA



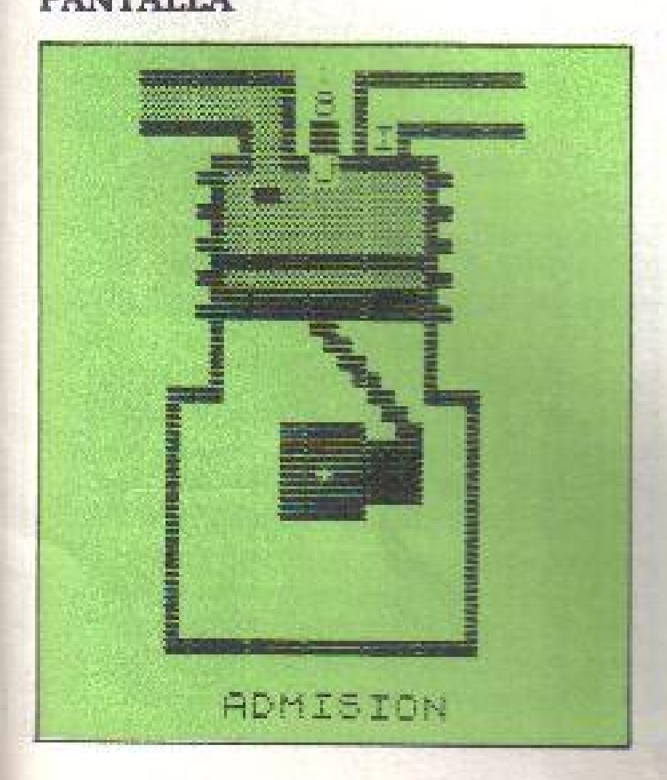
PANT.MOTOR: En este caso la subrutina en línea 700, juntamente con un programa en lenguaje de máquina agregado en línea 1, permite cargar imágenes que han sido grabadas en el casette a continuación de este programa. Luego de cargar el programa e iniciarlo con RUN, poner nuevamente en marcha el grabador y contestar "S", "ENTER" al principio y luego de cargar cada figura, en el intervalo que antecede a la próxima (mientras se carga una figura se ven las características barras blancas y negras en el televisor). Se cargan 9 figuras en total, las que presentadas luego en forma secuencial muestran el funcionamiento de un motor a explosión de 4 tiempos (Ideal para los padres que quieran explicar a sus hijos cómo funciona el motor del auto). Con una velocidad de 30 ó 40 pueden apreciarse claramente cada uno de los "tiempos".

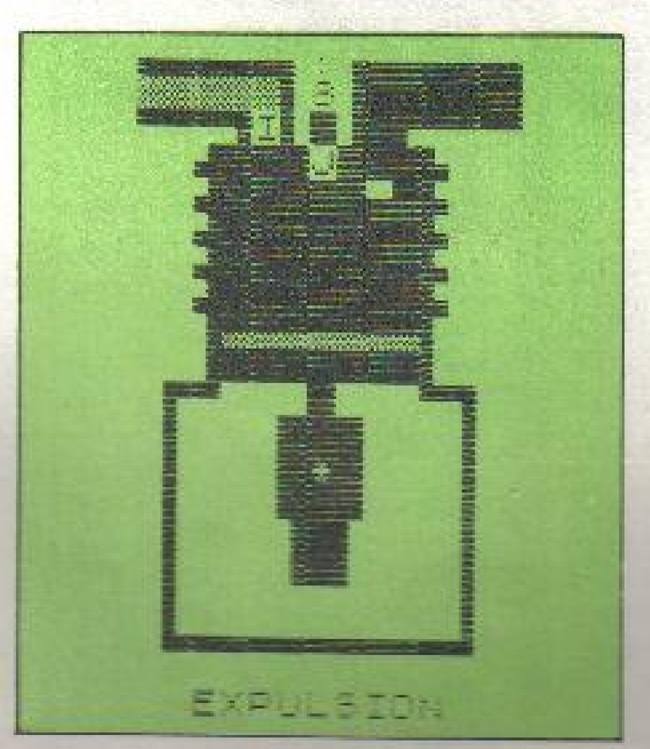
```
   1 REM 5URND GOSUB ??RNDACS 5,
??? FOR GOSUB ?£RND : GOSUB ?TA
N E£RND GOSUB ??RND : GOSUB ?TA
N LN SCROLL £RND?????VAL
RETURN PEEK COPY *K RAND
 RETURN *ACS ??3 PRINT ( PRINT 4
  RETURN ?K SLOW ZACS -14 THEN F
T "? COS /PEEK 6
   3 REM PROGRAMA PANT. EJEMPLO
DE APLICACION CON MOTOR DE 4 TIE
MPOS. E L VARETTI, LA PLATA 1985
'
  80 FAST
  90 DIM D$(704)
 100 POKE 16388,0
 110 POKE 16389,94
 120 LET T=PEEK 16388+256*PEEK 1
6389
 121 FOR I=1 TO 10
 123 LET P=T+(I-1)*800
 130 POKE (16443+2*I),P-256*INT
(P/256)
 140 POKE (16444+2*I),INT (P/256
)
 150 NEXT I
 160 CLS
 170 LET A=1
 180 PRINT "IMAGEN ";A;"?..(S/N
)"
 190 INPUT B$
 200 IF B$="N" THEN GOTO 400
 210 GOSUB 700
 280 POKE 16507,PEEK (16443+2*A)
 290 POKE 16508,PEEK (16444+2*A)
 300 RAND USR 16538
 310 LET A=A+1
 315 CLS
 316 LET D$=""
 320 IF A<=10 THEN GOTO 180
 400 CLS
 405 SLOW
 410 PRINT "QUE FIGURA?"
 420 INPUT A$
 421 PRINT "VELOCIDAD? (1-100)"
 422 INPUT B$
 423 LET B=VAL B$
 425 FOR I=1 TO LEN A$
 427 FOR J=1 TO B
 428 NEXT J
 440 POKE 16507,CODE A$(I)-28
 445 POKE 16508,0
 450 RAND USR 16514
 455 NEXT I
 457 IF CODE INKEY$=43 THEN GOTO
 410
 460 GOTO 425
 690 REM ***SUBRUTINA DE RECUPER
ACION DE IMAGENES.
 700 CLS
 720 RAND USR 16551
 730 PRINT D$
 770 RETURN
```

LISTADO EN LENGUAJE DE MAQUINA

```
16514=33        16515=59
16516=64        16517=237
16518=75        16519=123
16520=64        16521=203
16522=33        16523=9
16524=94        16525=35
16526=86        16527=235
16528=237       16529=91
16530=12        16531=64
16532=1         16533=25
16534=3         16535=237
16536=176       16537=201
16538=42        16539=12
16540=64        16541=237
16542=91        16543=123
16544=64        16545=1
16546=25        16547=3
16548=237       16549=175
16550=201       16551=205
16552=231       16553=2
16554=42        16555=16
16556=64        16557=35
16558=78        16559=35
16560=70        16561=35
16562=197       16563=30
16564=6         16565=219
16566=254       16567=211
16568=255       16569=23
16570=48        16571=249
16572=14        16573=148
16574=6         16575=26
16576=13        16577=219
16578=254       16579=23
16580=203       16581=121
16582=121       16583=56
16584=245       16585=16
16586=245       16587=32
16588=4         16589=251
16590=35        16591=48
16592=228       16593=63
16594=203       16595=22
16596=29        16597=32
16598=222       16599=193
16600=11        16601=120
16602=177       16603=200
16604=24        16605=211
16606=24
```

PANTALLA





LABERINTO

COMP: TS2068/CZ2000/TK90X
CONF: 48K
CLAS: ENT

Adaptado por: Juan Carlos Bruni

El programa original de este juego pertenece a una revista europea y está diseñado para ser corrido en una computadora ORIC 1.
Mi trabajo consistió en adaptarlo a la SINCLAIR, mejorándolo.
El programa comienza con la presentación, desde arriba, del laberinto, indicando dónde se encuentra el jugador.
A continuación aparece una habitación en 3 dimensiones en cuyas paredes hay puertas que le permitirán salir de esa habitación.
Las teclas para el desplazamiento son:
7=un paso hacia adelante. La puerta se abre y el jugador pasa a otra habitación.
5=Giro de 90° hacia la izquierda.
8=Giro de 90° hacia la derecha.
Para orientar al jugador, en la parte inferior de la pantalla aparece el rumbo, que refleja la orientación del jugador:

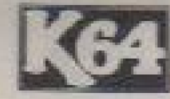
3 2 1 0

Es decir, que si el rumbo es 3, se está dirigiendo a la izquierda.
En la parte superior de la pantalla aparece el tiempo transcurrido desde el comienzo del juego.
En cualquier momento se puede ver el laberinto pulsando la letra "V"; pero ésto le restara puntos.
En el puntaje final intervienen: el tiempo que se tarda en salir del laberinto, y la cantidad de veces que se pulsó la letra "V".
La clave del programa está en la matriz de 3 dimensiones que contiene el laberinto codificado.

Las dos primeras dimensiones (X;Y) son las coordenadas de cada habitación.
La tercera dimensión (Z) identifica las 4 paredes de la habitación.
En una habitación existirá una puerta en alguna de las paredes si el valor de la matriz A(X,Y,Z) es 0. Si el valor de esta matriz es 1 denotará la ausencia de puerta.

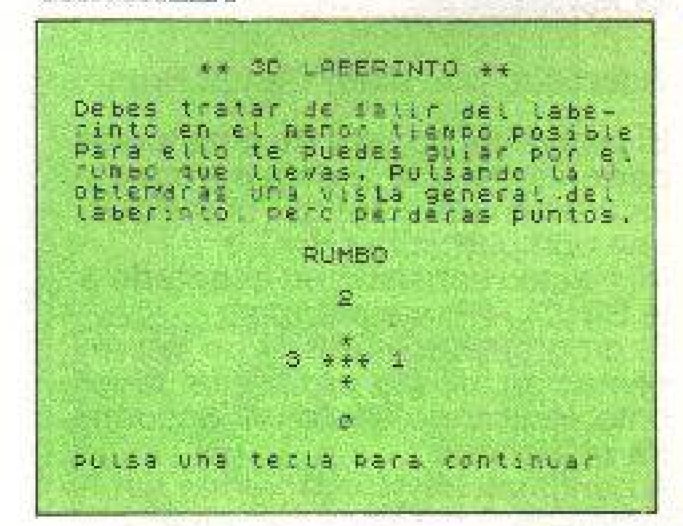
VARIABLES:

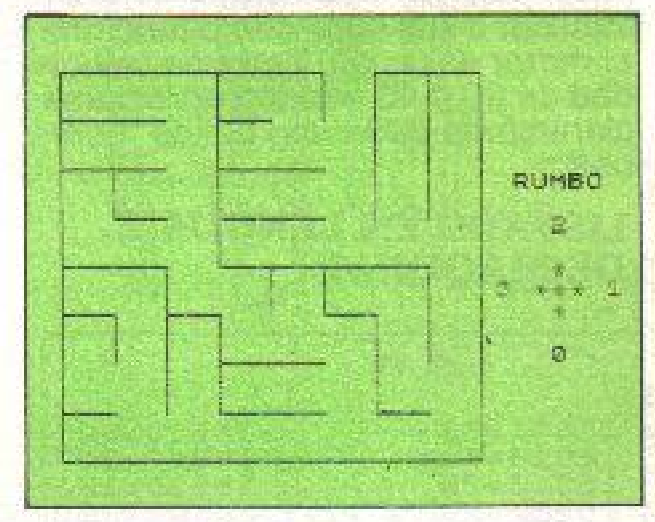
A(X,Y,Z)=matriz que contiene el laberinto codificado.
SC=puntaje final
HI=máximo puntaje
SC1°=puntaje provisorio
X,Y=coordenadas de la posición del jugador
DX,DY=valores que hay que sumar a las coordenadas para avanzar
X1,Y1=coordenadas provisionales
XX,YY=coordenadas de la puerta de salida
r=rumbo

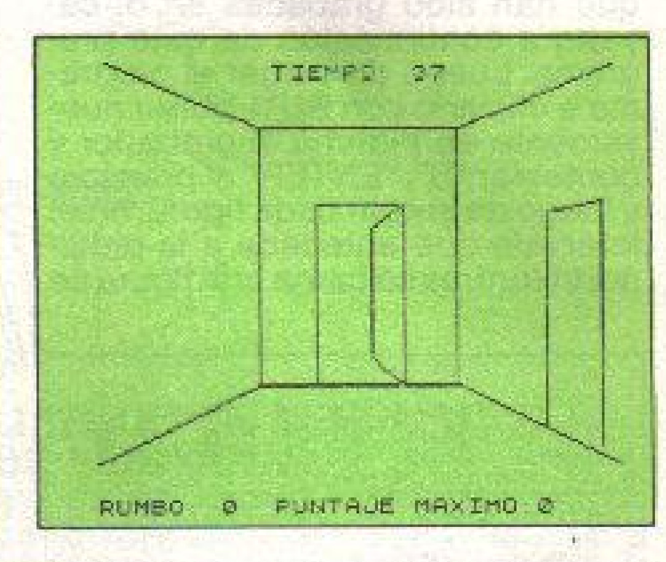


CONCURSO TRIMESTRAL

PANTALLA








```
   1 REM ** 3D LABERINTO **
   2 REM © JUAN CARLOS BRUNI
  10 DEF FN g()=INT ((PEEK 23672+256*PEEK 23673+65536*PEEK 23674)/50)
  20 POKE 23658,8
  30 PAPER 5: INK 2: BORDER 5: CLS
  40 GO SUB 8500
  70 LET hi=0
  80 LET sc1=50
  90 LET sc=0
 100 PRINT AT 15,5; FLASH 1;"LABERINTO DIBUJANDOSE"
 110 DIM a(8,8,4)
 120 FOR n=1 TO 8
 130 LET a(n,8,1)=1: LET a(n,1,3)=1
 140 LET a(1,n,4)=1: LET a(8,n,2)=1
 150 NEXT n
 155 RANDOMIZE
 160 LET a=RND: LET b=RND: LET c=RND
 170 IF a<0.5 THEN GO TO 210
 180 IF b<0.5 THEN GO TO 200
 190 LET a(1+INT (7*c),1,3)=2: LET xx=1+INT (7*c): LET yy=1: GO TO 240
 200 LET a(1+INT (7*c),8,1)=2: LET xx=1+INT (7*c): LET yy=8: GO TO 240
 210 IF b<0.5 THEN GO TO 230
 220 LET a(1,1+INT (7*c),4)=2: LET xx=1: LET yy=1+INT (7*c): GO TO 240
 230 LET a(8,1+INT (7*c),2)=2: LET xx=8: LET yy=1+INT (7*c)
 250 FOR m=1 TO 7
 260 FOR n=1 TO 7
 270 IF RND>0.5 THEN GO TO 310
 280 LET a(m,n,2)=1: LET a(m+1,n,4)=1
 290 LET a(m,n,1)=0: LET a(m,n+1,3)=0
 300 GO TO 330
 310 LET a(m,n,2)=0: LET a(m+1,n,4)=0
 320 LET a(m,n,1)=1: LET a(m,n+1,3)=1
 330 NEXT n
 340 NEXT m
 350 POKE 23672,0: POKE 23673,0: POKE 23674,0
 360 LET x=5: LET y=5: LET r=0: LET dx=0: LET dy=1
 370 LET a=1: LET b=2: LET c=4: GO TO 7000
 380 IF a(x,y,a)=1 THEN GO TO 430
 390 IF a(x,y,c)=1 THEN GO TO 420
 400 IF a(x,y,b)=1 THEN GO TO 2500
 410 GO TO 2000
 420 IF a(x,y,b)=1 THEN GO TO 3500
 425 IF a(x,y,b)=0 THEN GO TO 3000
 430 IF a(x,y,c)=1 THEN GO TO 450
 440 IF a(x,y,b)=1 THEN GO TO 4500
 445 IF a(x,y,b)=0 THEN GO TO 4000
 450 IF a(x,y,b)=1 THEN GO TO 5500
 460 IF a(x,y,b)=0 THEN GO TO 5000
 465 IF a(x,y,b)=2 THEN GO TO 3000
 470 PRINT AT 21,0;"RUMBO ";r;AT 21,10;"PUNTAJE MAXIMO ";hi;AT 0,10;"TIEMPO ";FN g()
 480 IF INKEY$="7" THEN GO SUB 1000: GO TO 620
 490 IF INKEY$="5" THEN GO SUB 1000: GO TO 540
 500 IF INKEY$="8" THEN GO SUB 1000: GO TO 560
 510 IF INKEY$="V" THEN GO TO 7000
 520 GO TO 470
 540 IF r<3 THEN LET r=r+1: GO TO 580
 545 IF r=3 THEN LET r=0: GO TO 580
 560 IF r>0 THEN LET r=r-1: GO TO 580
 565 IF r=0 THEN LET r=3
 580 IF r=0 THEN GO SUB 6000: GO TO 660
 590 IF r=1 THEN GO SUB 6100: GO TO 660
 600 IF r=2 THEN GO SUB 6200: GO TO 660
 610 LET dx=-1: LET dy=0: LET a=4: LET b=1: LET c=3: GO TO 660
 620 LET x1=x+dx: LET y1=dy+y
 630 IF a(x,y,a)=1 THEN LET x1=x: LET y1=y: GO TO 660
 640 IF a(x,y,a)=2 THEN GO SUB 8100: GO TO 9000
 650 LET x=x1: LET y=y1: GO SUB 8100
 660 GO TO 380
1000 FOR p=10 TO 30 STEP 5
1010 BEEP .01,p
1020 NEXT p
1040 RETURN
2000 GO SUB 8000
2010 PLOT 40,37: DRAW 0,87: DRAW -25,5: DRAW 0,-104
2020 PLOT 140,50: DRAW 0,70: DRAW -40,0: DRAW 0,-70
2030 PLOT 205,35: DRAW 0,82: DRAW 25,5: DRAW 0,-95
2040 GO TO 470
2500 GO SUB 8000
2510 PLOT 140,50: DRAW 0,70: DRAW -40,0: DRAW 0,-70
2520 PLOT 205,35: DRAW 0,82: DRAW 25,5: DRAW 0,-95
2550 GO TO 470
3000 GO SUB 8000
3010 PLOT 40,37: DRAW 0,87: DRAW -25,5: DRAW 0,-104
3020 PLOT 140,50: DRAW 0,70: DRAW -40,0: DRAW 0,-70
3030 GO TO 470
3500 GO SUB 8000
3510 PLOT 140,50: DRAW 0,70: DRAW -40,0: DRAW 0,-70
3520 GO TO 470
4000 GO SUB 8000
4010 PLOT 40,37: DRAW 0,87: DRAW -25,5: DRAW 0,-104
4020 PLOT 205,35: DRAW 0,82: DRAW 25,5: DRAW 0,-95
4030 GO TO 470
4500 GO SUB 8000
4510 PLOT 205,35: DRAW 0,82: DRAW 25,5: DRAW 0,-95
4520 GO TO 470
5000 GO SUB 8000
5010 PLOT 40,37: DRAW 0,87: DRAW -25,5: DRAW 0,-104
5020 GO TO 470
5500 GO SUB 8000
5510 GO TO 470
6000 LET dx=0: LET dy=1: LET a=1: LET b=2: LET c=4: RETURN
6100 LET dx=1: LET dy=0: LET a=2: LET b=3: LET c=1: RETURN
6200 LET dx=0: LET dy=-1: LET a=3: LET b=4: LET c=2: RETURN
7000 BORDER 4: PAPER 4: INK 2: CLS
7010 FOR m=1 TO 8
7020 FOR n=1 TO 8
7030 IF a(m,n,3)<>1 THEN GO TO 7050
7040 PLOT 23*(m-1)+10,172-18*(n-1): DRAW 23,0
7050 IF a(m,n,2)<>1 THEN GO TO 7070
7060 PLOT 23*(m-1)+33,172-18*(n-1): DRAW 0,-18
7070 NEXT n
7080 NEXT m
7090 FOR m=1 TO 8
7100 IF a(1,m,4)=2 THEN GO TO 7120
7110 PLOT 10,172-18*(m-1): DRAW 0,-18
7120 IF a(m,8,1)=2 THEN GO TO 7140
7130 PLOT 23*(m-1)+10,28: DRAW 23,0
7140 NEXT m
7150 PLOT FLASH 1;23*(x-1)+19,158-18*(y-1)
7160 PRINT AT 5,26;"RUMBO";AT 7,28;"2";AT 9,28;"*";AT 10,28;"*";AT 11,28;"*";AT 13,28;"0"
7170 PRINT AT 10,25;"3";AT 10,27;"*";AT 10,29;"*";AT 10,31;"1"
7180 LET sc1=sc1-10
7270 PAUSE 300: CLS : GO TO 380
8000 PAPER 5: BORDER 5: INK 1: CLS
8010 PLOT 0,20: DRAW 74,30: DRAW 0,100: DRAW -70,25
8020 PLOT 75,50: DRAW 89,0: DRAW 0,100: DRAW -90,0
8030 PLOT 239,20: DRAW -74,30: PLOT 165,150: DRAW 70,25
8040 RETURN
8100 PLOT 100,120: DRAW 40,0
8110 PLOT 140,120: DRAW -35,-10: DRAW 0,-50: DRAW 35,-10
8120 PLOT OVER 1;140,120: DRAW OVER 1;-35,-10: DRAW OVER 1;0,-50: DRAW OVER 1;35,-10
8130 PLOT 140,50: DRAW -25,5: DRAW 0,60: DRAW 25,5
8140 PLOT OVER 1;140,50: DRAW OVER 1;-25,5: DRAW OVER 1;0,60: DRAW OVER 1;25,5
8150 PLOT 140,50: DRAW -15,10: DRAW 0,50: DRAW 15,10
8170 RETURN
8500 PRINT AT 1,7;"** 3D LABERINTO **"
8510 PRINT AT 3,0;"Debes tratar de salir del labe- rinto en el menor tiempo posible"
8520 PRINT "Para ello te puedes guiar por elrumbo que llevas. Pulsando la V obtendras una vista general del laberinto, pero perderas puntos."
8530 PRINT AT 10,13;"RUMBO";AT 12,15;"2";AT 14,15;"*";AT 15,15;"*";AT 16,15;"*";AT 18,15;"0"
8540 PRINT AT 15,12;"3";AT 15,14;"*";AT 15,16;"*";AT 15,18;"1"
8550 PRINT AT 21,0; FLASH 1;"Pulsa una tecla para continuar": PAUSE 0
8560 CLS : PRINT AT 1,10;"COMANDOS";AT 3,0;"7 - un paso hacia adelante";AT 7,0;"8 - giro hacia la derecha";AT 9,0;"5 - giro hacia la izquierda";AT 11,0;"V - vista del laberinto"
8570 PRINT AT 21,0; FLASH 1;"Pulsa una tecla para empezar"
8600 PAUSE 0: RETURN
9000 CLS
9010 FOR N=1 TO 30
9020 BEEP .01,INT (RND*40)
9030 BORDER INT (RND*7)
9050 PRINT AT 10,8;"FELICITACIONES";AT 12,6;"HAS ENCONTRADO LA SALIDA"
9100 NEXT N
9110 LET t=FN g()
9120 LET sc=INT (10000/t+sc1)
9130 PRINT AT 17,5;"PUNTAJE ";sc
9140 IF sc>hi THEN LET hi=sc
9150 PRINT AT 19,5;"MAXIMO ";hi
9160 PRINT AT 21,0; FLASH 1;"Pulsa una tecla para otro juego"
9900 PAUSE 0: CLS : GO TO 80
9999 SAVE "laberinto" LINE 10
```

DISEÑADOR DE PANTALLAS

COMP: TS2068/CZ2000/TK90X
CONF: 48 K
CLAS: UTI

Autor: Juan Carlos Bruni

Este programa permite diseñar (dibujar) hasta 4 pantallas diferentes (laberintos, habitaciones, bases planetarias) y almacenarlas en la memoria.

Una vez completado el diseño, se borrará automáticamente el programa BASIC, dejando solamente las subrutinas de reposición de pantallas.

Solamente hay que agregar:
40 LOAD "" CODE,
para poder cargar luego las pantallas guardadas en la memoria. A continuación se podrá introducir el programa creado por el usuario.

La variable np se utilizará para especificar el número de pantalla que se desea reponer, y luego se llamará a la subrutina de reposición de pantalla mediante un GOSUB 9800.

Por ejemplo: al comenzar el juego se pondrá la pantalla 1:
LET np = 1 : GOSUB 9800.

Si sale por una puerta, pasará a la pantalla siguiente:
IF.....THEN LET np = 2 : GOSUB 9800.

La potencia de este programa reside en la brevísima subrutina en código de máquina que traslada el bloque de bytes desde el archivo de pantalla hasta la memoria, y viceversa.

De esta manera el pasaje de una pantalla a la otra es instantáneo. Por ejemplo, al salir por la puerta de una habitación, se presentará la otra habitación en forma inmediata.

Como la subrutina en código de máquina usa la memoria a partir de la dirección 65520, sólo se podrán usar los GDU (gráficos definidos por el usuario) hasta la letra S.

El menú permite seleccionar entre dibujar en pantalla de 22 x 32 (caracteres, letras, GDU) o en alta resolución (ploteo), borrar parte de lo dibujado y grabar en cassette la pantalla dibujada.

Si no se desean completar las 4 pantallas, se puede borrar el programa BASIC mediante la opción D del Menú.

En cualquier momento se puede acceder al menú, sin borrar el dibujo, pulsando la "M".

Si se está usando la opción "C" y se desea borrar algo, pulsar solamente la "B". Para retornar a la opción, pulsar "C". Lo mismo se puede decir de la opción "P" (borrar con "B"; retornar con "P").

RECOMENDACIONES:

En la programación del juego no podrán utilizarse las variables usadas por la subrutina: z, w, dfl, dfh, ml y mh.

La variable np, como se dijo, sirve para especificar el número de pantalla.

Una vez completadas las 4 pantallas se puede probar el programa de la siguiente manera:
1 FOR q = 1 TO 4 : LET np = q : GOSUB 9800 : NEXT q
2 STOP
RUN ENTER
De esta manera aparecerán las pantallas diseñadas.

VARIABLES

np = número de pantalla (1 a 4)
dx, dy = coordenadas del dibujo
dfl = byte menos significativo del archivo de pantalla.
dfh = byte más significativo del archivo de pantalla
ml = byte menos significativo de la zona de memoria donde se almacena la pantalla dibujada.
mh = byte más significativo de la zona de memoria donde se almacena la pantalla dibujada.

```
  10 REM DISENADOR DE PANTALLAS
  20 REM JUAN C. BRUNI - 1985 ©
  30 CLEAR 40790
  50 POKE 23658,8
  60 LET np=0
 100 GO SUB 8000
 110 PRINT AT 0,10;"M E N U";AT
5,5;"P= PLOTEA";AT 7,5;"C= DIBUJ
A CARACTERES";AT 9,5;"B= BORRA"
 120 PRINT AT 11,5;"A= ALMACENA
PANTALLAS";AT 13,5;"R= RECUPERA
PANTALLAS";AT 15,5;"D= BORRA PRO
GRAMA BASIC";AT 17,5;"S= GRABA E
N CINTA LA               PANTALL
A DIBUJADA"
 130 INPUT FLASH 1;"ELIJA OPCION
";k$
 140 CLS : GO TO 500
 490 INPUT FLASH 1;"ELIJA OPCION
";k$
 500 IF k$="P" THEN GO SUB 1000
 510 IF k$="C" THEN GO SUB 2000
 520 IF k$="A" THEN GO SUB 9000
 530 IF k$="R" THEN GO SUB 9700
 540 IF k$="S" THEN GO SUB 7000
 550 IF k$="D" THEN GO TO 9100
 560 GO TO 490
1000 REM plotea
1010 INPUT "COORDENADA X= ";dx;"
 Y= ";dy
1020 IF dx<0 OR dx>255 OR dy<0 O
R dy>175 THEN GO TO 1010
1030 PLOT dx,dy
1040 GO SUB 4200
1050 IF INKEY$="M" THEN RETURN
1070 IF INKEY$="B" THEN GO SUB 3
000
1080 GO TO 1030
2000 REM dibuja caracteres
2010 INPUT "QUE CARACTER? ";z$
2020 INPUT "COORDENADA X= ";dx;"
 Y= ";dy
2030 IF dx<0 OR dx>21 OR dy<0 OR
 dy>31 THEN GO TO 2010
2040 PRINT AT dx,dy;z$
2050 IF INKEY$="M" THEN RETURN
2060 IF INKEY$="B" THEN GO SUB 3
200
2070 GO SUB 4100
2080 GO TO 2040
3000 REM borra
3010 PLOT INVERSE 1;dx,dy: PLOT
dx,dy: PLOT INVERSE 1;dx,dy
3020 GO SUB 4200
3030 IF INKEY$="P" THEN RETURN
3040 GO TO 3010
3200 PRINT AT dx,dy;" ";AT dx,dy
;"*";AT dx,dy;" "
3210 GO SUB 4100
3220 IF INKEY$="C" THEN RETURN
3230 GO TO 3200
4000 REM control teclado
4100 LET dx=dx+(INKEY$="6" AND d
x<21)-(INKEY$="7" AND dx>0)
4110 LET dy=dy+(INKEY$="8" AND d
y<31)-(INKEY$="5" AND dy>0)
4120 RETURN
4200 LET dx=dx+(INKEY$="8" AND d
x<255)-(INKEY$="5" AND dx>0)
4210 LET dy=dy+(INKEY$="7" AND d
y<175)-(INKEY$="6" AND dy>0)
4220 RETURN
7000 REM graba pantalla
7010 SAVE "pantalla"SCREEN$
7020 RETURN
8000 REM instrucciones
8010 PRINT AT 0,4;"DISENADOR DE
PANTALLAS"
8020 PRINT '"UD. PUEDE DIBUJAR
HASTA CUATRO  PANTALLAS QUE SERA
N ALMACENADAS EN LA MEMORIA."
8030 PRINT '"LUEGO SE BORRARA AU
TOMATICAMENTEEL PROGRAMA BASIC Y
 ENTONCES UD.PODRA CREAR SU PROP
IO JUEGO."
8035 PRINT '"NO TOQUE LAS LINEAS
 10 A 60.     DEBE AGREGAR AL PRO
GRAMA         40 LOAD """" CODE"
8040 PRINT '"PARA REPONER LA PAN
TALLA DEBERA ASIGNARLE A LA VARI
ABLE np EL    NUMERO DE PANTALLA
DESEADO Y     LUEGO HACER UN ""GO
SUB 9800"""
8050 PRINT #0; FLASH 1;"PULSE UN
A TECLA"
8060 PAUSE 0: CLS : RETURN
9000 REM almacena pantallas
9010 LET np=np+1
9020 GO SUB 9900
9030 POKE 65521,dfl: POKE 65522,
dfh
9040 POKE 65524,ml: POKE 65525,m
h
9050 RANDOMIZE USR 65520
9060 CLS
9070 PRINT FLASH 1;"PANTALLA ";n
p;" ALMACENADA"
9080 IF np<4 THEN GO TO 9200
9090 PRINT "4 PANTALLAS ALMACENA
DAS"
9110 INPUT "BORRO EL PROGRAMA BA
SIC? (S/N)";p$
9120 IF p$="N" THEN GO TO 9200
9130 DELETE 100,9790
9200 PAUSE 100: CLS
9210 RETURN
9800 REM repone pantalla
9840 GO SUB 9900
9850 POKE 65524,dfl: POKE 65525,
dfh
9860 POKE 65521,ml: POKE 65522,m
h
9870 RANDOMIZE USR 65520
9890 RETURN
9900 RESTORE 9920
9910 FOR Z=65520 TO 65531: READ
W: POKE Z,W: NEXT Z
9920 DATA 33,0,0,17,0,0,1,255,23
,237,176,201
9930 LET dfl=0: LET dfh=64
9940 LET ml=87: LET mh=(159 AND
np=1)+(183 AND np=2)+(207 AND np
=3)+(231 AND np=4)
9970 RETURN
9998 SAVE "Disenador" LINE 1
9999 SAVE "pantallas"CODE 40791,
24580
```

PESCA

COMP: TS 2068
CONF: JOYSTICKS
CLAS: ENT

Autor: Carlos Hernán Marina

Es un juego que simula un real concurso de pesca.
Juegan dos jugadores al mismo tiempo. Hay otros tamaños de peces: de uno, dos y tres puntos. El juego tiene un límite puesto por un tiempo determinado, y cumplido ese tiempo se procede al recuento que da el ganador.
Se juega sólo con Joysticks. En el momento del pique, el jugador correspondiente deberá presionar el botón. Si esto se hace instantáneamente al ver el pique, se habrá clavado un pez. Luego de esto el pescado se recoge y al llegar al tope de su recorrido, tirando el joystick hacia adelante, se capturará el pez. Si esto no se hace con justeza el pez se cae perdiéndose.
Al final el ganador se define por los siguientes items:
Puntos: a jugador con más puntos ganará el partido. De haber empate se definirá por el número de piezas capturadas. De persistir la igualdad el ganador será aquel con mayor pieza promedio y de continuar iguales, triunfará el que haya tenido mayor porcentaje de eficiencia (piezas capturadas/piques). Después de estos rubros si no se establecen diferencias entre ambos, habrá definitivamente un empate.

PANTALLA

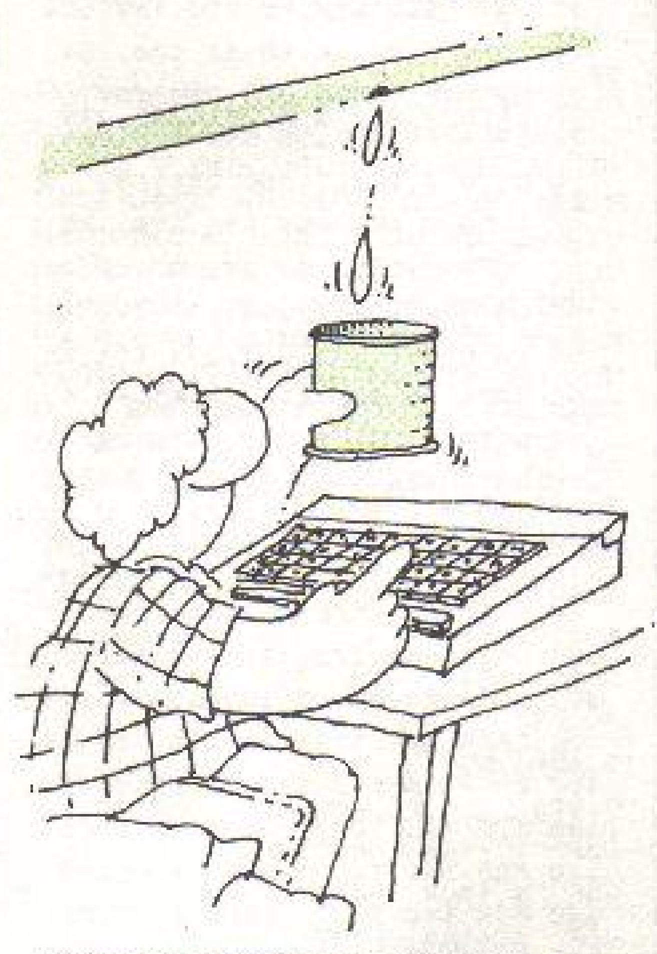



```
   1 POKE 23658,8: LET viv=0: CL
S
   2 GO SUB 9400: DATA -3,40,1,6
,3,10,10,5,10,2,7,-1,6,-6,0,-7,-
1,-12,-5,-7,-15,-7,-3,-2,0,-3,7,
-17,-2,-6,-7,-3,-10,3,-5,4
   3 DATA 3,3,27,5,3,-1,-1,-3,-2
,-2,-17,-4,-3,-2,-1,-5,2,1,14,0,
1,-2,-3,-8,-3,-2,-16,-2,-2,-10,1
5,2,4,-5,0,-5,-3,-6,-27,0,-4,3,4
,1,7,4,10,0,7,-2,3,-10,3,-3,-4,
-4,-8,0,-7,6,-7,-1,-3,-3,-1,-3,4
,-4,14,-3,8,-6,1,-6,-4,-10,-12,-
10,-12,4,-12,2,-4,3,5,0,5,2,8,-
1,2,-3,-2,-7,6,-3,10,2,5,5,0,4,-
5,3,-10,0,-6,2,1,5,4,5
   4 DATA 5,3,10,-1,5,-2,1,-5,2,
-5,-6,-3,-4,-3,-4,1,-4,-5,-2,-5,
0,-15,1,-4,3,-1,5,1,5,4,7,8,5,4,
7,2,7,1,3,2,-4,1,-4,5,-6,-5,-2,-
5,-7,-3,-6,-6,11,31,5,5,6,0,5,-3
,8,-30,0,-7,1,-3,4,-7,-11,-10,-1
,-8,-5,-5,0,18,-2,7,-5,0,-22,-18
,-12,-3,-9,1,-8,8
   5 DATA 52,82,60,60,24,24,60,1
02,20,52,85,125,125,50,82,60,60,
60,24,24,24,60,102,195,20,54,118
,94,126,126,126,126,126,126,126,
126,60,60,60,60
   6 DATA 60,60,24,24,24,60,102,
195,0,0,0,0,1,3,3,0,0,0,48,248,2
52,254,254,15,0,0,12,31,63,127,1
27,240,0,0,0,0,128,192,192,0
   7 FOR j=USR CHR$ 144 TO USR C
HR$ 153+7: READ a: POKE j,a: NEX
T
   8 IF viv<>0 THEN GO TO 10
   9 BORDER 1: GO SUB 1010: GO S
UB 79
  10 CLS : INK 0: PLOT 195,40
  11 DATA 6,15,0,5,3,5,-2,6,3,2,
-3,5,1,-2,50,0
  12 FOR g=0 TO 7
  13 READ a,b
  14 DRAW a,b
  15 NEXT g
  16 PLOT 231,94
  17 DATA 2,5,-1,11,-3,10,-4,2,-
1,-1,0,3,1,1,-2,5,-4,2,-4,-1,-1,
-2,0,-2,1,-1,0,-2,2,0,0,-1,2,0,-
1,-1,0,-2
  18 DATA 1,1,0,-1,-2,-1,5,1,2,3
,-2,-3,-5,-1,0,-4,-16,-7,-3,1,-2
,-2,2,-4,3,0,1,2,2,0,16,5,2,3,0,
2,-4,1,-1,-2
  19 FOR g=0 TO 38
  20 READ f,h
  21 DRAW f,h
  22 NEXT g
  23 DATA 3,-3,-1,-2,2,-2,-2,2,-
11,-2,-2,-2,-2,-17,-5,0,-1,1,-2,
-2,3,-2,10,-2,1,16,2,2,20,0
  24 PLOT 221,108: FOR g=0 TO 14
  25 READ q,w
  26 DRAW q,w
  27 NEXT g
  28 PLOT 220,127: PLOT 220,78
  29 DRAW -7,12
  30 PLOT 211,97: DRAW -2,7: PLO
T 206,109: DRAW -40,60: DRAW 39,
-61: PLOT 210,99: DRAW -4,5: PLO
T 221,78: DRAW -5,9
  74 LET i=0
  75 GO TO 87
  79 OVER 0: BORDER 1: INK 0: IF
 INKEY$="0" THEN GO TO 79
  81 CLS : PAPER 1: INK 7: BORDE
R 5: CLS : PRINT AT 7,10;"C - CO
MANDOS";AT 10,7;"ULTIMA LINEA -
JUEGO": BORDER 1
  82 IF INKEY$="C" OR INKEY$="c"
 THEN GO TO 9000
  83 IF INKEY$="Z" OR INKEY$="X"
 OR INKEY$="V" OR INKEY$="B" OR
INKEY$="N" OR INKEY$="M" OR INKE
Y$="z" OR INKEY$="x" OR INKEY$="
v" OR INKEY$="b" OR INKEY$="n" O
R INKEY$="m" THEN GO TO 86
  84 GO TO 82
  86 INK 0: PAPER 7: BORDER 1: R
ETURN
  87 LET e=0
  88 LET u=0
  89 LET s=0
  90 LET c=0
  91 LET p=0
  94 LET y=0
  95 LET d=0
  96 LET g=0
  97 LET x=0
  98 LET w=16
 100 PLOT 195,40
 101 DATA -135,0,-6,15,0,5,-3,5,
2,8,-3,2,3,5,-1,-2,-50,0
 102 FOR g=0 TO 6
 103 READ a,b
 104 DRAW a,b
 105 NEXT g
 106 LET a$="■"
 107 INK 6: PRINT AT 10,2;a$;AT
11,2;a$;AT 10,3;a$;AT 11,3;a$: I
NK 2;AT 10,29;a$;AT 11,29;a$;AT
10,28;a$;AT 11,28;a$: INK 0
 108 BEEP .1,12: PLOT 22,94
 109 DATA -2,5,1,11,3,10,4,2,1,-
1,0,3,-1,1,2,5,4,2,4,-1,1,-2,0,-
2,-1,-1,0,-2,-2,0,0,-1,-2,0,1,-1
,0,-2
 110 DATA -1,1,0,-1,1,-1,-5,1,-2
,3,2,-3,5,-1,0,-4,16,-7,3,1,2,-2
,-2,-4,-3,0,-1,2,-2,0,-16,5,-2,3
,0,2,4,1,1,-1
 111 FOR g=0 TO 38
 112 READ f,h
 115 DRAW f,h
 120 NEXT g
 130 PLOT 34,108
 132 DATA -3,-3,1,-2,-2,-2,2,2,1
1,-2,2,-2,2,-17,5,0,1,1,2,-2,-3,
-2,-10,-2,-1,16,-2,2,-20,0
 135 FOR g=0 TO 14
 139 LET tiempo=300
 140 READ q,w
 142 DRAW q,w
 143 NEXT g
 144 PLOT 35,127
 145 PLOT 32,78
 146 DRAW 7,12
 147 PLOT 44,97
 148 DRAW 2,7
 149 PLOT 50,109
 150 DRAW 40,62
 152 DRAW -39,-61
 154 PLOT 45,99
 156 DRAW 4,5
 158 PLOT 34,78
 160 DRAW 5,9
 161 OVER 0
 162 GO SUB 8000
 164 LET i=0
 165 LET g=0
 166 PAPER 7: INK 1: OVER 1: PRI
NT AT 17,0;I$;AT 17,24;O$: OVER
 173 LET w=16
 174 LET b=130
 175 BORDER 1: BORDER 5
 180 INK 0
 190 PAUSE 10
 200 IF STICK (2,2) OR STICK (2,
1) THEN GO TO 255
 210 PRINT AT 21,0; FLASH 1;"PRE
SIONAR EL BOTON PARA COMENZAR"
 250 GO TO 200
 255 PRINT AT 21,0; PAPER 1;"
"
 260 PAUSE 10: INK 0: PLOT 92,16
9: DRAW 0,-129
 261 OVER 1: INK 2: PRINT AT 16,
11;"*": OVER 0: INK 0
 262 IF i>0 THEN GO TO 275
 265 PAUSE 10: INK 0: PLOT 164,1
69: DRAW 0,-129
 270 OVER 1: INK 2: PRINT AT 16,
20;"*": OVER 0: INK 0
 275 LET i=i+1
 277 IF tiempo=0 THEN GO TO 1020
 280 LET k=INT (RND*102)
 290 LET l=INT (RND*12): IF l>10
 THEN GO TO 290
 291 LET kkk=0
 300 INK 0: PRINT AT 0,0;"puntos
:";y;"  ";AT 0,22;"puntos:";x;"
";AT 1,0;"piques:";e;AT 1,22;"p
iques:";u;AT 2,0;"piezas:";g;AT
2,22;"piezas:";p;AT 3,12;AT 0,12
;"TIEMPO ";AT 1,14;[illegible];"  "
 305 LET tiempo=INT (tiempo-.5)
 308 IF tiempo=0 THEN GO TO 1020
 309 IF STICK (2,1) THEN GO TO 3
12
 310 IF STICK (2,2) THEN GO TO 3
12
 311 GO TO 320
 312 PRINT FLASH 1; PAPER 3; INK
 7;AT 4,12;"[illegible]"
 313 BEEP .1,12
 314 GO TO 309
 320 PRINT FLASH 0;AT 4,12;"
  "
 321 IF k>93 AND k<=100 THEN GO
TO 400
 370 GO TO 280
 400 IF l>4 THEN LET r=20
 410 IF l<=4 THEN LET r=11
 420 PRINT AT 16,r;" "
 440 IF l<=4 THEN PAUSE 17.5: IF
 STICK (2,1) THEN GO TO 500
 442 IF l>4 THEN PAUSE 17.5: IF
STICK (2,2) THEN GO TO 500
 450 LET c=c+1
 455 IF tiempo=0 THEN GO TO 1000
 460 IF l>4 THEN LET u=u+1
 465 IF l<=4 THEN LET e=e+1
 470 IF l<=4 THEN GO TO 260
 475 GO TO 265
 500 BEEP .03,12: LET t=16
 505 IF STICK (2,1) OR STICK (2,
2) THEN GO TO 505
 510 IF k>=97 THEN GO TO 550
 520 IF k>=95 THEN GO TO 570
 530 LET a$=CHR$ 147
 532 LET b$=CHR$ 148
 535 LET c$=CHR$ 149
 540 LET j=3
 545 LET s=0
 547 GO TO 600
 550 LET a$=CHR$ 144
 553 LET b$=""
 554 LET c$=""
 560 LET j=1
 563 LET s=0
 565 GO TO 600
 570 LET a$=CHR$ 145
 575 LET b$=CHR$ 146
 580 LET c$=""
 585 LET s=0
 590 LET j=2
 600 IF l>4 THEN LET x=x+j
 610 IF l<=4 THEN LET y=y+j
 615 LET kkk=0
 646 LET t=t-1
 647 OVER 0: INK 0: PRINT AT t,r
;a$;AT t+1,r;b$;AT t+2,r;c$: OVE
R 0
```

GOTERA

COMP: TS 2068
CONF: Teclado/Joysticks
CLAS: ENT

Autor: Carlos Hernán Marina



```
 649 SOUND 6,6;7,7;8,16;9,16;10,
16;12,56;7,62;8,15;15,8
 650 IF L>4 THEN GO TO 656
 651 IF STICK (1,1)=1 THEN GO TO
 705
 655 GO TO 660
 656 IF STICK (1,2)=1 THEN GO TO
 705
 660 PRINT AT t,r;" ";AT t+1,r;"
 ";AT t+2,r;" "
 665 INK 1: OVER 0: PRINT AT 17,
r;"█";AT 18,r;"█": INK 0: OVER 0
 670 IF t=2 THEN GO TO 690
 680 IF tiempo=0 THEN GO TO 1000
 689 GO TO 620
 690 PRINT FLASH 0; INK 7;AT 4,4
;" ";AT 4,27;" ": SOUND 8,0;9,
0;10,0: IF L>4 THEN GO TO 694
 692 PAUSE 3: IF STICK (1,1)=1 T
HEN GO TO 800
 693 GO TO 695
 694 PAUSE 3: IF STICK (1,2)=1 T
HEN GO TO 800
 695 IF kkk=1 THEN GO TO 800
 700 LET t=t+1: INK 0
 706 PRINT FLASH 0; INK 7;AT 4,4
;" ";AT 4,27;" ": SOUND 8,0;9,
0;10,0: IF t>=16 THEN GO TO 750
 720 PRINT AT t,r;a$;AT t+1,r;b$
;AT t+2,r;c$
 730 PRINT AT t,r;" ";AT t+1,r;"
 ";AT t-2,r;" ": BEEP .01,12
 735 IF kkk=1 THEN GO TO 800
 740 GO TO 700
 751 IF L>4 THEN LET x=x-j
 752 IF L>4 THEN LET u=u+1
 755 IF L<=4 THEN LET y=y-j
 757 IF L<=4 THEN LET e=e+1
 758 LET c=c+1
 759 INK 1: OVER 0: PRINT AT 17,
r;"█";AT 18,r;"█": INK 0: OVER 0
 760 IF L>4 THEN GO TO 265
 762 GO TO 260
 800 IF kkk=0 THEN GO TO 808
 801 IF L>4 THEN LET x=x-j
 802 IF L<4 THEN LET y=y-j
 805 IF L>4 THEN LET u=u+1
 806 IF L<=4 THEN LET e=e+1
 807 LET c=c+1
 810 IF L>4 THEN LET w=w+1
 822 IF L<=4 THEN LET d=d+1
 830 IF w=31 THEN LET w=18
 840 IF d=14 THEN LET d=0
 850 OVER 1: PAPER 6: INK 1: IF
L<=4 THEN PRINT AT 19,d;a$;AT 20
,d;b$;AT 21,d;c$: BEEP .01,1: OV
ER 0
 860 OVER 1: PAPER 7: INK 1: IF
L>4 THEN PRINT AT 19,w;a$;AT 20,
w;b$;AT 21,w;c$: BEEP .01,1: OVE
R 0
 865 INK 0: PAPER 7
 870 IF L>4 THEN LET p=p+1
 880 IF L<=4 THEN LET g=g+1
 890 IF tiempo=0 THEN GO TO 1000
 900 OVER 0: INK 0
 950 IF L>4 THEN GO TO 265
 953 GO TO 260
1000 SOUND 8,0;9,0;10,0: CLS
1005 GO TO 1100
1010 IF INKEY$="0" THEN GO TO 10
10
1011 OVER 0: INK 0
1015 CLS : PAPER 1: INK 7: BORDE
R 6: CLS : PRINT AT 8,1;"NOMBRE
JUGADOR IZQ.DE PANTALLA";AT 11,5
;"no mas de 8 letras": INPUT LIN
E I$
1020 PRINT AT 8,1;"NOMBRE JUGADO
R DER.DE PANTALLA": INPUT LINE O
$: RETURN
1100 CLS
1200 PRINT AT 1,1;I$;AT 1,20;O$;
AT 4,1;"PUNTOS: ";y;AT 6,1;"PIEZA
S: ";g;AT 4,20;"PUNTOS: ";x;AT 6,2
0;"PIEZAS: ";p;AT 18,2;"en el con
curso hubo ";c;" piques"
1240 IF g=0 THEN LET g=1
1245 IF p=0 THEN LET p=1
1250 PRINT AT 10,1;"pieza/p.: ";IN
T (y/g);AT 10,20;"pieza/p.: ";INT
(x/p)
1270 PRINT AT 8,1;"piques: ";e;AT
 8,20;"piques: ";u
1275 IF (y/g)=0 THEN LET g=0
1280 IF (x/p)=0 THEN LET p=0
1290 IF e=0 THEN LET e=1
1295 IF u=0 THEN LET u=1
1300 PRINT AT 13,1;"efic.: ";INT
(g*100/e);" %";AT 13,20;"efic.: "
;INT (p*100/u);" %"
1310 IF y<>x THEN GO TO 1400
1320 IF g<>p THEN GO TO 1430
1330 IF INT (p*100/u)<>INT (g*10
0/e) THEN GO TO 1470
1335 IF g=0 THEN LET g=1
1336 IF p=0 THEN LET p=1
1350 IF (y/g)<>(x/p) THEN GO TO
1500
1360 PRINT AT 16,9;"E M P A T E"
1400 IF y>x THEN PRINT AT 16,5;"
GANADOR: ";I$
1410 IF x>y THEN PRINT AT 16,5;"
GANADOR: ";O$
1420 GO TO 1900
1430 IF g>p THEN PRINT AT 16,5;"
GANADOR: ";I$
1440 IF p>g THEN PRINT AT 16,5;"
GANADOR: ";O$
1450 GO TO 1900
1470 IF (g*100/e)>(p*100/u) THEN
 PRINT AT 16,5;"GANADOR: ";I$
1480 IF (p*100/u)>(g*100/e) THEN
 PRINT AT 16,5;"GANADOR: ";O$
1490 GO TO 1900
1500 IF (y/g)>(x/p) THEN PRINT A
T 16,5;"GANADOR ";I$
1510 IF (x/p)>(y/g) THEN PRINT A
T 16,5;"GANADOR: ";O$
1511 FOR A=0 TO 50: BEEP .03,A:
BEEP .0006,A: NEXT A
1900 PRINT AT 19,0;"Para revanch
a X";AT 20,0;"Para empezar de nu
evo E";AT 21,0;"Para copiar C":
BEEP .1,1
2000 IF INKEY$="x" OR INKEY$="X"
 THEN GO TO 2150: BEEP .1,1
2010 IF INKEY$="e" OR INKEY$="E"
 THEN RUN
2050 IF INKEY$="c" OR INKEY$="C"
 THEN GO TO 2200
2100 GO TO 2000
2150 RESTORE : LET v;v=10: GO TO
 2
2200 PRINT AT 19,0;"


                      ": COPY : GO
 TO 1900
2300 STOP
8011 BEEP .1,1
8020 OVER 1: INK 1: PRINT AT 17,
2;[redacted] OVER 0
8050 FOR q=12 TO 16
8055 BEEP .005,1
8070 INK 4: PRINT AT q,0;"[redacted]
"
8090 NEXT q
8100 OVER 1: PRINT AT 16,6;"█";A
T 15,6;"█";AT 14,6;"█";AT 13,6;"
█";AT 12,6;"█";AT 16,7;"█": OVER
 0
8110 FOR q=12 TO 16
8111 BEEP .005,1
8130 PRINT AT q,26;"[redacted]"
8140 NEXT q
8150 OVER 1: PRINT AT 16,25;"█";
AT 15,25;"█";AT 14,25;"█";AT 13,
25;"█";AT 12,25;"█";AT 16,24;"█"
: OVER 0
8160 INK 1: PRINT AT 5,3;CHR$ 15
0+CHR$ 151: INK 2: PRINT AT 5,27
;CHR$ 152+CHR$ 153: OVER 0
8200 RETURN
9000 CLS : PAPER 1: INK 6
9001 IF INKEY$="c" OR INKEY$="C"
 THEN GO TO 9001
9005 OVER 0
9010 GO SUB 9200: PRINT AT 10,5;
"HACIA ARRIBA : RECOGE";AT 12,5;
"BOTON ROJO   : CLAVA"
9015 PAUSE 0
9020 CLS
9040 INK 0: PAPER 7: BORDER 1: G
O TO 10
9200 PAPER 2: INK 7: BORDER 5: P
RINT AT 1,0;"
400 COMAt
9300 RETURN
9400 CLS : BORDER 1: PAPER 3: IN
K 7: CLS
9490 PLOT 50,50
9500 FOR g=0 TO 17
9510 READ a: READ b
9520 DRAW a,b
9530 NEXT g
9540 CIRCLE 71,94,5
9545 PLOT 65,109
9550 FOR g=0 TO 21
9560 READ a,b
9570 DRAW a,b
9575 NEXT g
9580 PLOT 118,112
9585 FOR g=0 TO 19
9590 READ a,b
9600 DRAW a,b
9610 NEXT g
9620 PLOT 93,71
9625 FOR g=0 TO 11
9627 READ a,b
9640 DRAW a,b
9650 NEXT g
9655 PLOT 108,95
9656 FOR g=1 TO 2
9660 READ a,b
9665 DRAW a,b
9670 NEXT g
9675 PLOT 147,109
9680 FOR g=0 TO 43
9685 READ a,b
9690 DRAW a,b
9695 NEXT g
9700 PLOT 138,91: DRAW 1,4: DRAW
 3,4
9710 PRINT FLASH 1; INK 0; PAPER
 6;AT 1,4;"** por HERNAN MARINA
**"
9720 FOR A=0 TO 1
9721 FOR b=0 TO 7
9722 BEEP .05,b+20: BEEP .0008,b
+30
9725 BORDER b
9726 NEXT b: NEXT a
9740 PAPER 7: INK 0: BORDER 1: C
LS
9750 RETURN
```

El juego consiste en atrapar la mayor cantidad de gotas posibles que caigan desde el techo.

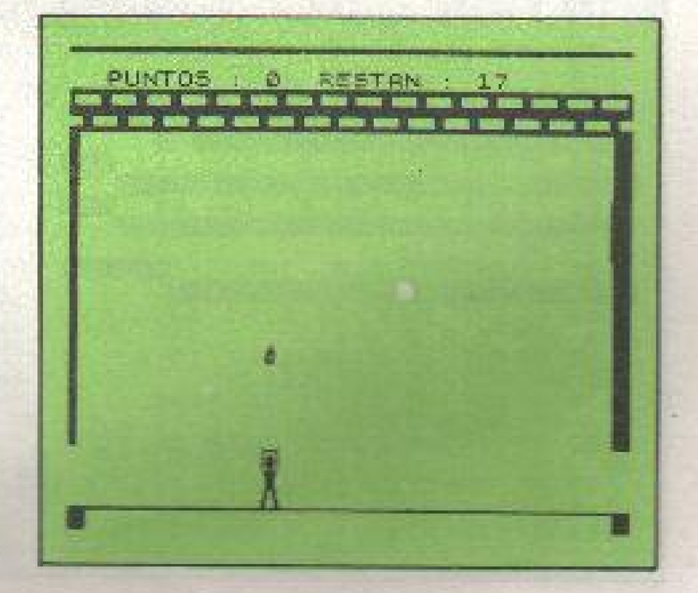
A medida que las gotas no sean capturadas se irá llenando de agua un depósito, cuando esté lleno acabará el juego. (Se llena a las 17 gotas).

Puede jugarse con joysticks o con el teclado, indistintamente.

En el juego constan los controles y sus efectos.

Sobre el margen derecho se observará una compuerta que se irá abriendo a medida que se llene el depósito, y cuando esto suceda se abrirá totalmente provocando el final del juego.

PANTALLA





```
   1
   2
   3 REM  **PROGRAMA  GOTERA**
   4
   5
   6
   7
   8
  50 CLS
  55 DATA 128,255 66,153,189,189
,189,153
  60 DATA 8,24,24,60,52,100,124,
56
  68 DATA 0,0,0,0,0,0,255,255
  70 DATA 0,0,0,0,0,255,255,255
  72 DATA 0,0,0,255,255,255,255,
255
  74 DATA 0,0,255,255,255,255,25
5,255
  76 DATA 0,255,255,255,255,255,
255,255
  80 DATA 146,73,34,146,73,42,44
,253
  85 DATA 255,255,192,192,192,19
2,255,255
  90 DATA 255,255,3,3,3,3,255,25
5
  92 DATA 90,126,126,60,60,60,60
,60
  94 DATA 24,24,24,24,24,24,24,2
55
  96 DATA 102,102,102,102,195,19
5,195,255
  97 DATA 71,31,159,62,108,175,1
87,159
  98 DATA 63,191,59,123,191,62,4
7,15
  99 DATA 252,122,250,216,220,18
6,178,249
 100 DATA 255,253,191,255,239,25
5,251,191
 101 DATA 24,36,68,153,189,189,1
62,153
 102 DATA 129,129,129,153,189,18
9,189,153
 103 DATA 64,112,124,62,62,63,12
7,255
 105 FOR t=0 TO 7: READ a: POKE
USR "a"+t,a: NEXT t
 110 FOR t=0 TO 7: READ a: POKE
USR "b"+t,a: NEXT t
 120 FOR t=0 TO 7: READ a: POKE
USR "c"+t,a: NEXT t
 125 FOR t=0 TO 7: READ a: POKE
USR "d"+t,a: NEXT t
 130 FOR t=0 TO 7: READ a: POKE
USR "e"+t,a: NEXT t
 135 FOR t=0 TO 7: READ a: POKE
USR "f"+t,a: NEXT t
 140 FOR t=0 TO 7: READ a: POKE
USR "g"+t,a: NEXT t
 145 FOR t=0 TO 7: READ a: POKE
USR "h"+t,a: NEXT t
 146 FOR t=0 TO 7: READ a: POKE
USR "i"+t,a: NEXT t
 147 FOR t=0 TO 7: READ a: POKE
USR "j"+t,a: NEXT t
 150 FOR t=0 TO 7: READ a: POKE
USR "k"+t,a: NEXT t
 155 FOR t=0 TO 7: READ a: POKE
USR "l"+t,a: NEXT t
 160 FOR t=0 TO 7: READ a: POKE
USR "m"+t,a: NEXT t
 162 FOR t=0 TO 7: READ a: POKE
USR "n"+t,a: NEXT t
 163 FOR t=0 TO 7: READ a: POKE
USR "o"+t,a: NEXT t
 164 FOR t=0 TO 7: READ a: POKE
USR "p"+t,a: NEXT t
 165 FOR t=0 TO 7: READ a: POKE
USR "q"+t,a: NEXT t
 166 FOR t=0 TO 7: READ a: POKE
USR "r"+t,a: NEXT t
 167 FOR t=0 TO 7: READ a: POKE
USR "s"+t,a: NEXT t
 168 FOR t=0 TO 7: READ a: POKE
USR "t"+t,a: NEXT t
 170 BORDER 3: INK 0: PAPER 5: C
LS
 172 LET ee=-1
 174 FOR f=21 TO 1 STEP -3
 176 LET ee=ee+1
 178 PRINT AT f,0; INK ee;"[redacted]"
 180 PRINT AT f-1,0; INK ee;"[redacted]"
 182 PRINT AT f-2,0; INK ee;"[redacted]"
 184 NEXT f
 186 PRINT AT 0,0; INK 7;"[redacted]"
 188 PRINT INVERSE 1;AT 10,6;"*
* G O T E R A * *"
 189 PRINT INVERSE 1;AT 15,4;"C
-COMANDOS//RESTO-JUEGO"
 190 PRINT AT 21,7; INVERSE 1;"P
OR HERNAN MARINA"
 192 IF STICK (2,1) THEN GO TO 1
98
 193 IF INKEY$="c" OR INKEY$="C"
 THEN GO SUB 9500
 194 IF INKEY$<>"" THEN GO TO 19
8
 196 GO TO 192
 198 IF STICK (2,1) OR INKEY$<>"
" THEN GO TO 198
 199 BORDER 5: PAPER 7: CLS
 200 FOR f=1 TO 16
 205 IF f=5 OR f=6 OR f=7 THEN G
O TO 217
 215 PRINT AT f+1,0;"";AT f+1,3
1;""
 216 GO TO 220
 217 PRINT AT f+1,0;"";AT f+1,3
1; INK 3;""
 220 NEXT f
 225 PLOT 247,103: DRAW 0,24
 230 PRINT AT 1,0; INK 0; PAPER
6;""
 240 PRINT AT 2,0; INK 0; PAPER
6;""
 300 PRINT AT 0,0; PAPER 3;"
 310 OVER 1
 320 PRINT AT 20,0;"
 "
 330 OVER 0
 500 LET x=15
 505 PRINT INK 0;AT 18,x;""
 506 PRINT INK 0;AT 19,x;""
 507 PRINT INK 0;AT 20,x;""
 510 LET y=3
 520 LET t=0: LET c$="1"
 530 LET gc=0
 540 LET ga=0
 560 LET a$="
 "
 590 PRINT AT 21,0; INK 2;"";AT
 21,31;""
 600 IF INKEY$<>"" THEN GO TO 90
0
 610 IF STICK (2,1) THEN GO TO 9
00
 620 GO TO 600
 900 LET xx=INT (RND*100)-40: IF
 xx>28 OR xx<1 THEN GO TO 900
 910 LET pun=ga*10-gc
 930 LET on=17-gc
 940 PRINT AT 0,0; INK 7; PAPER
3;"  PUNTOS   ";pun;"   ";"RESTAN
  ";on;" "
 950 LET b$=""
 960 LET d$=" "
1000 LET x=x+(INKEY$="8")-(INKEY
$="5")
1005 IF STICK (1,1)=8 THEN LET x
=x+1
1007 IF STICK (1,1)=4 THEN LET x
=x-1
1010 IF x<1 THEN LET x=1
1020 IF x>28 THEN LET x=28
1030 IF x/2=INT (x/2) THEN LET c
$=""
1035 IF x/2<>INT (x/2) THEN LET
c$=""
1036 IF STICK (2,1) OR INKEY$="2
" THEN GO TO 1042
1040 GO TO 1050
1042 IF STICK (2,1) OR INKEY$="0
" THEN GO TO 1042
1044 IF STICK (2,1) OR INKEY$="0
" THEN GO TO 1050
1045 PRINT AT 5,12; FLASH 1; PAP
ER 3; INK 7;"DETENIDO": BEEP .25
,35
1046 GO TO 1044
1050 PRINT AT 5,12;"        ": P
RINT AT 18,x-1;" "
1051 PRINT AT 19,x-1;" "
1052 PRINT AT 20,x-1;"_";c$;"_"
1055 LET y=y+1
1062 IF xx=x AND y=18 THEN GO TO
 5000
1070 IF y=21 THEN LET b$=""
1075 IF y=21 AND x<>xx THEN LET
d$="_"
1077 IF y=21 AND x=xx THEN LET d
$=c$
1078 IF y=20 AND x=xx THEN LET d
$=""
1080 PRINT AT y,xx; INK 3;b$;AT
y-1,xx; INK 0;d$
1090 IF y=21 THEN GO TO 7000
1500 GO TO 1000
5000 PRINT INK 3;AT y-1,x;""
5005 PAUSE 3
5010 PRINT AT y-1,xx;" "
5050 LET ga=ga+1
5070 BEEP .05,28: BEEP .05,20
5100 LET y=3: GO TO 900
7005 LET y=3
7010 LET gc=gc+1
7020 BEEP .01,50: BEEP .03,40
7050 IF gc=2 THEN LET a$="______
"
7060 IF gc=4 THEN LET a$="______
"
7070 IF gc=6 THEN LET a$="______
"
7080 IF gc=8 THEN LET a$="[redacted]"
7090 IF gc=10 THEN LET a$="[redacted]"
7100 IF gc=12 THEN LET a$="[redacted]"
7110 IF gc=14 THEN LET a$="[redacted]"
7120 IF gc=16 THEN LET a$="[redacted]"
7130 PRINT INK 0;AT 21,1;a$
7140 PLOT INK 7;247,103+gc
7150 IF gc=17 THEN GO TO 8000
7160 GO TO 900
8000 IF INKEY$<>"" THEN GO TO 80
00
8010 PAUSE 80: INK 3
8050 PRINT AT 6,27;"[redacted]"
8055 PRINT AT 7,26;"[redacted]"
8060 PRINT AT 8,26;"[redacted]"
8125 LET h=7
8130 FOR f=9 TO 21
8140 PRINT AT f,26; INK 3;"[redacted]"
8145 SOUND 6,6;7,7;8,16;9,16;12,
56;13,8
8150 NEXT f
8170 FOR f=21 TO 7 STEP -1
8180 PRINT AT f,1;"______________________"
8190 PRINT AT f,1;"______________________"
8200 PRINT AT f,1;"______________________"
8210 PRINT AT f,1;"______________________"
8220 PRINT AT f,1;"______________________"
8230 PRINT AT f,1;"[redacted]"
8240 PRINT AT f,1;"[redacted]"
8250 PRINT AT f,1;"[redacted]"
8260 SOUND 6,6;7,7;8,16;9,16;12,
56;13,8
8265 LET h=h+1
8267 IF h>=f THEN LET h=7
8270 PRINT AT h,27;"[redacted]"
8271 IF h=7 THEN GO TO 8290
8272 PRINT AT h-1,27;"[redacted]"
8290 NEXT f
8295 PRINT AT 5,1;"
8297 PRINT AT 7,27;"[redacted]"
8300 SOUND 8,0;9,0;10,0
8310 BEEP .05,27
8320 PRINT AT 6,15; INK 3;""
8325 PAUSE 8
8330 FOR f=1 TO 31
8340 IF f/2=INT (f/2) THEN LET k
$=""
8345 IF f/2<>INT (f/2) THEN LET
k$=""
8346 BEEP .005,56: BEEP .01,40
8347 BEEP .005,12: BEEP .01,60
8352 PRINT AT 6,15; INK 0;k$
8355 IF f<20 THEN GO TO 8369
8358 PRINT AT 6,f-19; INK 0;" "
8369 NEXT f
8370 PRINT AT 6,13;""
8372 PAUSE 5: PRINT AT 6,13;" "
8377 PAUSE 30
8380 PRINT AT 6,15; INK 3;""
8385 PAUSE 6
8390 PRINT AT 6,15; INK 3;" "
8400 BEEP .05,50
8999 INK 0: PAUSE 60
9000 BORDER 0: PAPER 7: INK 0: C
LS
9010 PRINT AT 7,10;"PUNTOS ";p
un
9020 PRINT AT 9,6;"GOTAS ATRAPAD
AS  ";ga
9030 PRINT AT 11,6;"GOTAS LANZAD
AS  ";ga+gc
9100 INK 0: PAPER 7
9110 IF STICK (2,1) OR INKEY$<>"
" THEN GO TO 9140
9120 GO TO 9110
9140 GO TO 170
9500 PAPER 3: BORDER 3: INK 7: C
LS
9510 PRINT AT 9,10;"J-JOYSTICK";
AT 11,10;"T-TECLAS"
9520 IF INKEY$="J" OR INKEY$="j"
 THEN GO TO 9600
9530 IF INKEY$="t".OR INKEY$="T"
 THEN GO TO 9700
9540 GO TO 9520
9600 CLS
9610 PRINT AT 7,7;"HACIA IZQ.=IZ
QUIERDA";AT 9,7;"HACIA DER.=DERE
CHA";AT 11,7;"BOTON=DETIENE EL J
UEGO"
9620 IF STICK (2,1) THEN GO TO 1
70
9625 IF INKEY$<>"" THEN GO TO 17
0
9630 GO TO 9620
9700 IF INKEY$<>"" THEN GO TO 97
00
9705 PRINT AT 7,7;"5=HACIA DEREC
HA";AT 9,7;"8=HACIA IZQUIERDA";A
T 11,7;"0=DETIENE EL JUEGO"
9710 IF INKEY$<>"" THEN GO TO 17
0
9720 GO TO 9710
```

RAPIDEZ DE REFLEJOS

COMP: C-64, Vic-20, PET/Commodore
CONF: 1 K
CLAS: ENT

Autor: Dr. Roberto Mercader

Pruebe cómo andan sus reflejos con su 64. Siga las instrucciones del programa y si su tiempo de reacción es mayor que 0,7 segundos ¡por favor no salga con su auto!



CAPITALES

COMP: C-64, VIC-20, PET/Commodore
CONF: 1,5 K
CLAS: EDU

Autor: Dr. Roberto Mercader

Por medio de este programa la computadora elige al azar una provincia y pregunta por su capital. Este juego educativo acelera grandemente el tiempo de aprendizaje. Puede usarse también con cualquier lista de nombres asociados. Para ello basta modificar en las sentencias DATA los pares de nombres correspondientes y cambiar en las sentencias 100 y 180 los números 23 por el número total de sentencias DATA de la nueva lista. Muy instructivo para niños en edad escolar.




```
100 REM RAPIDEZ DE REFLEJOS. K64-JUN-85
110 PRINT"[PARA MEDIR LA RAPIDEZ DE SUS REFLEJOS"
120 PRINT"XAPRIETE CUALQUIER TECLA Y LUEGO ESPERE"
130 PRINT"XA QUE APAREZCA EL SONIDO."
140 PRINT"XCUANDO LO ESCUCHE, APRIETE NUEVAMENTE "
150 PRINT"XCUALQUIER TECLA TAN RAPIDO COMO PUEDA."
160 PRINT"XBUENA SUERTE!"
170 GOTO 180
180 PRINT"XAPRIETE CUALQUIER TECLA"
190 GET A$
200 IF A$=""THEN190
210 PRINT
220 PRINT"XXATENTO AL SONIDO!■"
230 PRINT
240 Q=TI+INT(60+301*RND(0))
250 GET A$
260 IFA$<>""THEN 460
270 IF TI<Q THEN 250
280 POKE 54295,0
290 POKE 54296,15
300 POKE 54278,240
310 POKE 54277,16
320 POKE 54273,100
330 POKE 54276,33
340 X=TI
350 GET A$
360 IFA$=""THEN350
370 R=TI
380 POKE 54276,0
390 POKE 54296,0
400 PRINT"SU TIEMPO DE REACCION ES "
410 Z$=STR$((R-X)/60):Z$=RIGHT$(Z$,LEN(Z$)-1)
420 IF(R-X)/60<1THEN PRINT"X      0"+Z$;"   SEGUNDOS.":GOTO440
430 PRINT"X       "Z$;"   SEGUNDOS."
440 INPUT"XOTRA VEZ (DEF/N)";R$:IFR$<>"N"THEN 180
450 GOTO 480
460 PRINT"XNO SEA TAN APURADO!"
470 GOTO 180
480 END
```

```
100 POKE53280,12:POKE53281,1:POKE646,0
110 DIMP$(23),C$(23),K(23):FORP=1TO23:READP$(P),C$(P):NEXT
120 B=0:T=0:PRINT"XXCOMO QUIERE JUGAR
130 PRINT"X 1)NOMBRANDO LA CAPITAL.
140 PRINT"X 2)NOMBRANDO LA PROVINCIA.
150 PRINT"X 3)AMBAS.
160 POKE198,0:WAIT198,1:GETA$:A=VAL(A$):G=A
170 T=T+1:IFT>15THEN300
180 Q=INT(RND(1)*23)+1:IFK(Q)=1THEN180
190 ONGGOTO200,210:ONINT(RND(1)*2)+1GOTO200,210
200 PRINT"XCUAL ES LA CAPITAL DE";P$(Q)"?":R$=C$(Q):GOTO220
210 PRINT"XEN QUE PROVINCIA ESTA";C$(Q)"?":R$=P$(Q)
220 A$="":INPUTA$:IFA$=""THEN220
230 IFA$<>R$THEN260
240 PRINT"XCORRECTO!":B=B+1:K(Q)=1
250 GOTO280
260 PRINT"XEQUIVOCADO
270 PRINT"LA RESPUESTA CORRECTA ES":PRINT"X"R$"■
280 IF T<15THEN PRINT"XX"B"CORRECTAS SOBRE"T"PREGUNTAS
290 GOTO170
300 FORI=1TO1000:NEXT
310 PRINT"XXHA HA SIDO"B"RESPUESTAS CORRECTAS SOBRE   UN TOTAL DE"T-1"PREGUNTAS
320 PRINT"XXCALIFICACION ";INT((10*B/(T-1))+.5)"PUNTOS.
330 INPUT"XXOTRA VEZ (SI/NO)";H$
340 IFH$="SI"THENFORI=1TO23:K(I)=0:NEXT:GOTO120
350 END
360 DATABUENOS AIRES,LA PLATA
370 DATACATAMARCA,CATAMARCA
380 DATACORDOBA,CORDOBA
390 DATACORRIENTES,CORRIENTES
400 DATACHACO,RESISTENCIA
410 DATACHUBUT,RAWSON
420 DATAENTRE RIOS,PARANA
430 DATAFORMOSA,FORMOSA
440 DATAJUJUY,SAN SALVADOR DE JUJUY
450 DATALA PAMPA,SANTA ROSA
460 DATALA RIOJA,LA RIOJA
470 DATAMENDOZA,MENDOZA
480 DATAMISIONES,POSADAS
490 DATANEUQUEN,NEUQUEN
500 DATARIO NEGRO,VIEDMA
510 DATASALTA,SALTA
520 DATASAN JUAN,SAN JUAN
530 DATASAN LUIS,SAN LUIS
540 DATASANTA CRUZ,RIO GALLEGOS
550 DATASANTA FE,SANTA FE
560 DATASANTIAGO DEL ESTERO,SANTIAGO DEL ESTERO
570 DATATUCUMAN,SAN MIGUEL DE TUCUMAN
580 DATATIERRA DEL FUEGO,USHUAIA
```



QUE ES UN PERIFERICO



Llamamos periféricos a los elementos que permiten introducir o sacar datos del computador. Existe una gran cantidad de ellos, algunos de usos muy específicos, que permiten una aplicación cada vez más variada de los computadores ya que es a través de los periféricos que un equipo se convierte.

Debido a la importancia que tienen los periféricos dentro del tema computación, trataremos a la vez de explicar su funcionamiento, dar sus características principales, de forma tal que todos puedan saber en qué fijarse para elegir entre los periféricos, cuáles se pueden conectar al computador, y elegir el que mejor se adapte a sus necesidades.

Una forma de clasificar los periféricos según la forma en que transfieren los datos al computador central es la siguiente:

PERIFERICOS DE ENTRADA:

TECLADO
MOUSE
JOYSTICK
LAPIZ ELECTRONICO
DIGITALIZADOR

PERIFERICOS DE SALIDA:

MONITOR
IMPRESORA
PLOTTER

PERIFERICOS DE ENTRADA/SALIDA:

GRABADOR DE CASSETTES
LECTO/GRABADOR DE DISCOS FLEXIBLES (DRIVE)
LECTO/GRABADOR DE DISCO RIGIDO (HARD DISK)
MICRODRIVES
TERMINAL DE VIDEO

- Veremos en detalle los principales periféricos.

TECLADO: En general es difícil separar este periférico de la CPU propiamente dicha, por ser el medio de entrada por excelencia sin embargo debe estar claro que es un elemento totalmente separable, por lo menos funcionalmente.
Sus características principales son:
Tipo de tecla (membrana-switch)
Cantidad de teclas
Teclado numérico separado
Teclas de funciones
Acentos y ñ
Diseño ergonométrico

- Estos son los principales atributos que tenemos que mirar cuando elegimos un teclado.

MONITOR: Otro elemento indispensable en un equipo de computación. A veces se lo sustituye con un televisor, pero las características principales son las mismas. Además de la obvia diferenciación entre monitores de color y blanco y negro, deben separarse los de alta y baja resolución; siendo la resolución la capacidad de dibujar dos puntos cercanos en la pantalla y poder diferenciar uno del otro. Por este motivo solamente en los de alta resolución pueden dibujarse gráficos bien definidos.
En los de baja resolución se suele hablar de su capacidad en número de caracteres por línea y de líneas por pantalla, dándose esta característica como = 80 x 24 que significa 80 caracteres por línea y 24 líneas por pantalla, sin embargo esta característica no es una particularidad del monitor propiamente dicho, sino que es un atributo del controlador de video que está en la computadora. Por lo tanto cuando elegimos un computador debemos fijarnos cuántos caracteres por línea y cuántas líneas por pantalla despliega.
En los de alta resolución la definición es más difícil y se da generalmente por pixeles que puede manejar.
Siendo el pixel la mínima unidad de graficación, o sea "el punto", por lo que se da la cantidad de puntos que se pueden dibujar en la pantalla por ejemplo = 16.000 pixeles.
Otros atributos del monitor son, su tamaño dado en pulgadas (9", 10", 12", etc). Esta dimensión se refiere a la **diagonal** de la pantalla y no a un lado como a veces se piensa.
En la próxima nota veremos con más detalle las impresoras y sus características principales.

INGENIERO NINO MORENO

En esta sección atendemos todas aquellas consultas y sugerencias que nuestros lectores deseen realizar. Para ello sólo debe dirigirse a esta redacción, sección "Consultas"

INTERFACE 1

Me gustaría saber si la interface "ZX 1" para la conexión de "Microdrives" de la computadora ZX Spectrum es compatible con la computadora TS 2068. Además tengo la duda de si un modem se conecta directamente en las conexiones de la computadora Spectrum o hace falta alguna interface. Una vez conectado, la Spectrum puede comunicarse con una Texas TI99/4A; con un Apple o con una IBM, para acceder a información, intercambio de programas, etc.
Desde ya cuento con su grata colaboración.
Néstor Hugo López
Cabanillas
Lomas de Zamora

K-64:
La interface 1 no es compatible con la TS 2068; además por lo que sabemos no estarán disponibles a la venta por algunos meses. Por otro lado, disponiendo de una ZX Spectrum con la Interface 1, se le puede conectar cualquier modem que tenga entrada RS-232 y su Software adecuado para controlarlo. También existen Modems que no necesitan interfaces y se conectan directamente en el conector posterior. En cualquiera de los casos, es posible conectarse con cualquier otra computadora que tenga conectado un Modem que trabaje en la misma norma, (Bell 103 ó CCITT), como así también comunicarse con Bases de datos internacionales.

HALLAZGOS

Es gratificante comprar una nueva revista de computación y descubrir que ha superado a todas las conocidas. ¿Por qué? Es muy sencillo. En general, todas las revistas que traen programas tienen defectos de impresión y bien sabemos que confundir un punto con una coma puede resultar catastrófico.
Por otra parte los usuarios de Sinclair y sus similares hemos encontrado que una gran parte de los programas están elaborados para nuestros computadores.
En una palabra, el gasto se justifica plenamente.
No quisiera seguir con halagos pues resultaría excesivo y todos los que nos dedicamos a la computación sabemos que son más importantes los hechos concretos que las palabras. Es por ello que les envío una modesta colaboración en las hojas adjuntas.
Creo que sería una idea interesante si los lectores tuviésemos una sección para comentar nuestros hallazgos. Yo lanzo la propuesta, usted será el encargado de concretarla si ello es posible.
Esperando que su revista siga avanzando como lo ha hecho en estos dos números, lo saluda atentamente.

Capacidad operativa del Sistema Basic Sinclair

Las posibilidades de programación con el sistema Basic Sinclair son muy amplias. Existen una serie de "trucos" que son muy útiles y no figuran en el manual que acompaña a la máquina. He rescatado alguno de ellos de distintos programas que han llegado a mis manos. Espero que a los lectores de su revista les sean tan prácticos como lo fueron para mí.

1- GOTO CONDICIONADO

Todos sabemos que nuestras Sinclair no poseen la sentencia ON GOTO. El manual propone su sustitución por GOTO nº * A, pero esto sólo es útil cuando el número de línea al que queremos saltar es mútiplo de A. Una solución que evita expresiones matemáticas complejas con la consiguiente demora en su hallazgo es la siguiente:

Ej:
GOTO (1000 AND B=1) + (1800 AND B=2)
El mando será enviado a 1000 cuando B sea igual a 1 ó será enviado a 1800 cuando B sea igual a 2.

2- FOR Y PRINT AT CONDICIONADO

a- FOR CONDICIONADO:
Ej:
FOR I=3*(0=1)+10*(0=2) TO 5*(0=)+0*(0=2) STEP 1−2*(0=2)
Si 0 es igual a 1 hará:
FOR I=3 TO 5 STEP 1
Si 0 es igual a 2 hará:
FOR I=10 TO 0 STEP −1
b- PRINT AT CONDICIONADO
Ej:
PRINT AT 10,3*(0=1)+22*(0=2);A
Si 0 es igual a 1 hará:
PRINT AT 10,3;A
Si 0 es igual a 2 hará:
PRINT AT 10,22;A
Este tipo de instrucciones es muy útil para juegos que requieren el movimiento de piezas o figuras.

3- IF CON DETENCION

A veces queremos imprimir un mensaje y luego detener la ejecución de un programa si se cumple una determinada con-

dición. (ej: finalización de un juego). Una forma de hacerlo es la siguiente:
IF Z=30 THEN PRINT "UD. GANA";T
T debe ser una variable no definida en el programa. De esta manera se genera una condición de error y el programa se detiene.

4- IFs CONCECUTIVOS

Ej:
IF A=1 THEN IF B=2 THEN PRINT "UD. PIERDE"
Sólo se imprimirá el mensaje si A=1 y B=2

5- INPUT DE CADENAS

En algún tipo de programas es útil el siguiente tipo de instrucción:
INPUT A$(1,3 TO 5)
Su ejecución produce el almacenamiento de los caracteres ingresados en la cadena A$ en los lugares 3,4 y 5 de la misma.
Este tipo de estructuras permite el almacenamiento de largas cadenas conteniendo una información completa de un ítem.
Por ejemplo: Artículo, precio, cantidad, etc. Luego esta información puede ser tabulada con instrucciones como la siguiente:
PRINT AT 0,0;A$(1,TO 3); AT 0,5;A$(1,3 TO 5);....
Por supuesto que no hay que olvidar el dimensionamiento previo.

Creo que estas habilidades del Basic Sinclair pueden resultar de interés a los lectores, que como yo, tratamos de obtener el máximo rendimiento de nuestras máquinas. Queda pues, a su criterio, su publicación.

Gustavo Kleiman
Godoy Cruz Mendoza

K-64:

Agradecemos sus elogios y buenos deseos. Por otra parte, bienvenida su colaboración! Esperamos que sea contagioso.

BASIC "STANDARD"

Encuentro a vuestra publicación demasiado demasiado sectaria; considero que en EE.UU. y Europa es posible limitarse a una sola marca, dado la cantidad de publicaciones que cubren todo el espectro de micros. Pero estamos en Argentina, y aún no existe una publicación alternativa con vuestro perfil.
No les pido abandonar a Sinclair o TI, no. Sólo que bajen del pedestal y se dignen explicar —cuando corresponda— las particularidades de cada programa que apunten a un determinado "dialecto" o arquitectura de memoria. Ya sé que éso puede resultar terriblemente pesado. Pero hay una alternativa; un par de artículos describiendo lo "no estandar" de Sinclair y TI en cuanto a sus "basic", y una pequeña aclaración acerca de donde tienen y cómo se organizan sus "memorias pantalla". Lo no cubierto en estas notas podrá completarse con breves aclaraciones a los programas cuando sea necesario.
También es posible y para demostrarlo los mando al libro "Juegos para Ordenador" de T. Hartnell, Ed. Anaya, escribir programas razonablemente normalizados, escritos sin "peek" ni "poke".
No pretendo iniciar una polémica sólo que no se olviden de quienes también compramos la revista y tenemos otros computadores...
Les saluda uno de vuestros lectores no Sinclair.
Juan Carlos Rossi
1203 Buenos Aires

K 64

Acusamos recibo de su carta agradeciendo sus sugerencias.
El propósito de K64 es cubrir el mercado de las microcomputadoras más utilizadas.
Tenemos programas de Sinclair, de TI99, de Commodore y de Microdigital.
En cuanto al pedestal no es nuestra filosofía; nuestro objetivo es prestar servicio a los usuarios e ir incrementando la relación con ellos.
La definición de un Basic "Standard", está muy polemizada y referirse al de Microsoft sería caminar hacia atrás. Por otro lado, los programas "estandarizados" como los que Ud. sugiere (poniendo como ejemplo el libro de Hartnell), son faltos de "vida" y no permiten la explotación de las virtudes particulares de cada máquina.
Lo que sí haremos es explicar un poco más a fondo estas máquinas más populares, para que usuarios como Ud. puedan adaptar o aprovechar los programas publicados.
Estamos interesados en conocer cuántos usuarios de otros modelos estarían interesados en ver publicados programas originales, sin necesidad de corregirlos, como así también saber de la existencia de clubes de usuarios, para que se conozcan entre sí a través de K64, y puedan aumentar su intercambio por todo el país y hasta el exterior.
Lamentamos que al final, no nos haya mencionado de qué marca se trata su computadora, una Apple, IBM PC, o tal vez una Altair.

LS 10

TU CONEXION CON LA MUSICA.

RADIO De Plata LS 10

1030AM/95.1FM ESTEREO.



El único límite del microcomputador MICRODIGITAL TK85 es "SU IMAGINACION".

OPCIONALES:

• Joystick (órgano de comando externo), para juegos de video • Generador de sonido
• Expansor de memoria de 64 Kbytes RAM • Impresora directa • Programador de EPROM • Interface para conexión de impresora profesional (paralela).

MICRODIGITAL TK-85

Importa y distribuye: ARVOC s.a.i.c.f.i.
Avda. DIAZ VELEZ 4147 (1200) Capital Tel.: 981-1980/9212

GARANTIA: 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electrónica, fotografía y librerías.